LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

DISSERTATION

Dennis J. Miller, Captain, USAF

AFIT/DS/ENP/93-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DS/ENP/93-02

1

LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

DISSERTATION

Dennis J. Miller, Captain, USAF

AFIT/DS/ENP/93-02

Approved for public release; distribution unlimited

AFIT/DS/ENP/93-02

LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

Dennis J. Miller, B.S., M.S.

Captain, USAF

Accepted:

Kirk A Mathews 15 June 93
Kirk A Mathews, Chairman

Denis E. Beller 15 June 93
Denis E. Beller, Member

Dennis W Quinn 15 JUNE 93
Dennis W. Quinn, Member

Charles Bridgman 15 June 93
Charles J. Bridgman, Dean's Representative

Accepted:

Dean, School of Engineering

AFIT/DS/ENP/93-02

LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

DISSERTATION

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Dennis J. Miller, B.S., M.S.
Captain, USAF

June 1993

Approved for public release; distribution unlimited

Preface

The purpose of this study was to develop a two-dimensional general triangle linear characteristic spatial quadrature for discrete ordinates neutral particle transport. This quadrature will increase the analyst's capability to model problems with complex geometry. This method should easily extend to three dimensions.

This work involved extensive theoretical development and computer modeling. Triangular cell conservation relationships were derived, a general triangle discrete ordinates computer algorithm was developed and implemented, and the triangle linear characteristic spatial quadrature was derived and transformed into computational form.

In this development I have had a great deal of help from others. I would first and foremost like to thank my faculty advisor, Dr. Kirk A. Mathews, for his help and great insight and for suggesting the problem and basic approach. I would like to thank my committee for their time and advice during this effort. I would also like to thank Wolfram Research Inc. whose development of the software package Mathematica was invaluable to this research. Finally, I would like to thank my wife, Laurie, for her compassion on those long nights when I was a slave to my computer terminal and my children, Jeremy and Christopher, for their sacrifice when dad just couldn't be there.

Dennis J. Miller

Table of Contents

Page

Preface ii

List of Figures vi

List of Tables ix

I. Introduction 1

A. Background 2

1. Neutral Particle Boltzmann Transport Equation 2
2. Discrete Ordinates 4

B. Problem Statement 6

C. Scope 6

D. Sequence of Presentation 7

II. Transport with Rectangular Spatial Cells 9

A. Discretizing in Angle for the Transport Equation 9

B. Solution Techniques for the Boltzmann Equation 10

1. Monte Carlo 10
2. Discrete Ordinates 11

C. Conservation in Two-Dimensional Cartesian Geometry 13

1. Cell-balance for the Rectangular Cell 13
2. X-moment Conservation 15
3. Y-moment Conservation 17

D. Angular Quadratures 17

1. Level Symmetric Quadratures 17
2. Product Quadratures 18

E. Spatial Quadratures for Rectangular Cells 18

1. Rectangular Step Method 19
2. Rectangle Diamond Difference 20
3. Rectangle Step Characteristic 21
4. Rectangle Linear Characteristic 23

5. Other Rectangle Quadratures 27

III. General Triangle Spatial Quadrature Development 29

A. The Local Coordinate System 29

B. Conservation Equations 35

1. Case 0 Cell-Balance 35
2. Case 0 U-moment Conservation 37
3. Case 0 V-moment Conservation 38
4. Orientation for First Order Edge-Moments 39

C. Rotation/Translation between Coordinate Frames 39

1. Cell Zeroth Order (Average) Moments 40
2. First Order Moments 41
3. Splitting of Edge Zeroth and First Moments 43
4. Assembly of Edge Zeroth and First Moments 46

D. Case 0 Spatial Quadratures 46

1. Case 0 Step Method 47
2. Case 0 Step Characteristic. 47
3. Case 0 Linear Characteristic 48

E. Triangle Mesh Development and Refinement 55

IV. General Triangle Algorithm (TRISN) 57

A. The SNXY Rectangle Discrete Ordinates Algorithm 57

B. The TRISN Triangle Discrete Ordinates Algorithm 59

V. Testing 70

A. Benchmarking TRISN with Lathrop's Problem 70

B. Convergence Rate of Triangle Linear Characteristic 76

C. Sensitivity of Linear Characteristic to Triangle Mesh Variation 78

D. Test Case 1: Source Cylinder with Annular Segment Reflectors 86

1. Spatial Mesh Considerations for the Partially Reflected Cylinder Problem. 89
2. Fully Closed Configuration 95
3. Half Open Configuration 96

4. Fully Open Configuration 97

E. Test Case 2: The Vacuum Duct 98

2. Narrow Vacuum Duct 99

3. Wide Vacuum Duct 108

F. Execution Speed 114

G. Summary of Observations 115

VI. Conclusions and Recommendations 117

Bibliography 121

Vita 123

List of Figures

Figure Page

1. The Rectangle Unit Cell 15

2. An Arbitrary Triangle 30

3. Case 0 Triangle. 32

4. Case 1 Triangle. 33

5. Case 2 Triangle. 33

6. Case 0 Triangle Edge-Moment Orientation. 37

7. Triangle Edge-moment Direction Convention. 40

8. Input Edge-moment Splitting 45

9. General Triangle Mesh Refinement 56

10. Lathrop's Benchmark Problem for TRISN. 71

11. Initial Triangle Mesh for Lathrop's Problem 73

12. Lathrop's Problem Benchmark Results for TRISN vs. SNXY 74

13. Top Edge Partial Current for Benchmark Problem 75

14. Top Edge Current Relative Difference of Squares vs. Triangles for Benchmark Problem 76

15. Average Scalar Flux Convergence Rate for Rectangle/Triangle Linear Characteristic 78

16. Mesh Sensitivity Test Set for the Sensitivity Test Problem for 0.25 Unit Random Variation 80

17. Relative Difference in Scalar Flux for Triangle Linear Characteristic 81

18. Top Edge Current Relative Difference From Square Calculations for Sensitivity Problem for 0.25 Unit Random Variation in Node Placement 83

19. Mesh Sensitivity Test Set for the Sensitivity Test Problem for 0.50 Unit Random Variation 84

20. Top Edge Current Relative Difference for Sensitivity Problem for 0.50 Unit Random Variation in Node Placement 85

21. Fully Closed Partially Reflected Cylinder Problem 87

22. Half Open Partially Reflected Cylinder Problem 88

23. Fully Open Partially Reflected Cylinder Problem 89

24. Fully Open Partially Reflected Cylinder Problem (80 triangle mesh) 90

25. Fully Open Partially Reflected Cylinder Problem (204 square mesh) 91

26. Half Open Partially Reflected Cylinder Problem (80 triangle mesh) 92

27. Half Open Partially Reflected Cylinder Problem (208 square mesh) 93

28. Fully Closed Partially Reflected Cylinder Problem (80 triangle mesh) 93

29. Fully Closed Partially Reflected Cylinder Problem (208 square mesh) 94

30. Fully Closed Configuration Source Average Scalar Flux 96

31. Half Open Configuration Source Average Scalar Flux 97

32. Fully Open Configuration Source Average Scalar Flux 98

33. Narrow Vacuum Duct Problem 100

34. Narrow Vacuum Duct (4 triangle mesh) 101

35. Narrow Vacuum Duct (6 square mesh) 101

36. Narrow Vacuum Duct (12 triangle mesh) 102

37. Narrow Vacuum Duct (72 square mesh) 102

38. Relative Error Narrow Vacuum Duct Problem (scatter case) 105

39. Narrow Vacuum Duct 10/12 Triangle Mesh Comparison 106

40. Top Edge Current Narrow Vacuum Duct (scatter case)................ 106

41. Relative Error Narrow Vacuum Duct Problem (absorber case) 108

42. Wide Vacuum Duct Problem.. 109

43. Top Edge Current Wide Vacuum Duct Problem (scatter case)....... 111

44. Relative Error Wide Vacuum Duct Problem (scatter case)............ 112

45. Top Edge Current Wide Vacuum Duct Problem (absorber case).... 114

List of Tables

Table Page

1. Top Edge Current for 0.25 Unit Random Variation in Node Placement for the Sensitivity Test Case 82

2. Top Edge Current for 0.50 Unit Random Variation in Node Placement for the Sensitivity Test Case 83

3. Areas for Regions of Various Meshes for the Fully Open Partially Reflected Cylinder Problem 91

4. Spatial Mesh Parameters for Partially Reflected Cylinder Problem 94

5. Spatial Mesh Parameters for Narrow Vacuum Duct Problem........ 103

6. Average Top Edge Narrow Current Vacuum Duct Problem (scatter case) 104

7. Average Top Edge Current Narrow Vacuum Duct Problem (absorber case) 107

8. Average Top Edge Current Wide Vacuum Duct Problem (scatter cast) 110

9. Average Top Edge Current Wide Vacuum Duct Problem (scatter case) 113

AFIT/DS/ENP/93-02

Abstract

A new spatial quadrature for tl.e discrete ordinates method for an arbitrary mesh of triangular cells is derived and compared to the rectangular cell linear characteristic spatial quadrature. The triangular mesh is more flexible, allowing curved surfaces and off-axis angles to be approximated with many fewer spatial cells. The method is consistently more accurate than rectangular linear characteristic on example problems tested here. Arbitrary orientation and size of the triangles allows non-patterned meshes to be developed which appears to ameliorate numerical diffusion. Linear characteristic and arbitrary triangle meshes are a desirable alternative to linear characteristic and square meshes on many problems.

The general triangle linear characteristic spatial quadrature achieves nearly the same asymptotic rate of convergence (to the same result) as rectangular linear characteristic on Lathrop's problem. Mesh sensitivity measurements indicate that large variations in triangle cell vertex locations produce less than 1.0 percent variation in results. Test cases included a rectangular region with a diagonal vacuum duct, and cylindrical source region with various rotated rings of annular segmented reflectors. The triangle linear characteristic quadrature is more cost effective on these problems achieving a relative error of less than 1.0 percent with a factor of anywhere from three to more than a hundred fewer spatial cells, with less than three times the computational cost per spatial cell. This spatial cell savings should increase the domain of practical problems for which the discrete ordinates method is usable.

LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

I. Introduction

Rectangle based spatial quadratures are used widely because of the simplicity of rectangle mesh generation. This simplicity allows a great deal of flexibility in spatial quadrature development. However, problem shapes, such as curvilinear boundaries, provide a great deal of difficulty for rectangular meshes. Rectangle mesh refinement requirements may be determined by geometry concerns and not by particle transport. Curvilinear spatial quadratures are an alternative for some problems but the complexity involved in their development greatly limits the types of approximations that can be used. Arbitrary triangle meshes can approximate nearly any curvilinear boundary with many fewer cells than rectangle meshes and with less complexity and more general applicability than curvilinear methods.

The research reported here has developed and tested a new numerical method for solving the Boltzmann neutral particle transport equation: the general triangle linear characteristic spatial quadrature (two-dimensional Cartesian geometry).

A. Background

1. Neutral Particle Boltzmann Transport Equation

Neutral particle transport is controlled by a balance equation known as the Boltzmann transport equation. This equation gives the neutral particle angular flux, ψ, at any point in seven-dimensional phase space. The angular flux is a function of position $(\vec{r})$, energy (E) or speed (v), time (t), and direction of travel $(\hat{\Omega})$. The Boltzmann equation is an integro-differential equation and has the following form

$$L\psi(\vec{r},\hat{\Omega},E,t) = H\psi(\vec{r},\hat{\Omega}',E',t) + S_{EXT}(\vec{r},\hat{\Omega},E,t), \tag{1}$$

where L is known as the streaming and collision operator and accounts for particle losses from the phase element. The L operator accounts for the time dependent change in flux, the loss of particles due to streaming, and scatter/absorption losses. The operator L is given by

$$L = \left[\frac{1}{v}\frac{\partial}{\partial t} + \hat{\Omega}\cdot\vec{\nabla} + \sigma_t(\vec{r},E)\right]. \tag{2}$$

The integral source operator H accounts for scattering sources into the phase element, fission sources, and flux independent sources. H is given by

$$H = \left[\int dE'\int d\hat{\Omega}'\,\sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})\right] + \left[\chi(E)\sum_i(1-\beta_i)\int d\Omega'\int dE'\nu\sigma_f(\vec{r},E')\right] \tag{3}$$

where

$\hat{\Omega}$ is a unit vector in the direction of particle motion.

$\sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})dE\,d\Omega$ is the probability per unit path length that particles at position $\vec{r}$ with energy E'

traveling in direction $\hat{\Omega}'$ will scatter into energies in dE about E and into directions of motion in $d\Omega$ about $\hat{\Omega}$.

$\chi(E)\nu\sigma_f(\vec{r},E')dE\,d\Omega$ is the probable of number particles emitted at position $\vec{r}$ with energies in dE about E and directions in $d\Omega$ about $\hat{\Omega}$ per unit path length of travel of particles of energy E'.

$S_{EXT}(\vec{r},\hat{\Omega},E,t)dE\,d\Omega$ is the emission rate density of particles from sources that are independent of the flux distribution emitted at position $\vec{r}$ with energies in dE about E and directions in $d\Omega$ about $\hat{\Omega}$.

Spatial quadrature development does not require the complexity of equation (1). The Boltzmann transport equation is simplified using the following assumptions:

1) the energy distribution can be represented by one energy group (monoenergetic),

2) the problem is in steady state,

3) particle scatter and sources are isotropic,

4) two-dimensional geometry, and

5) no fission sources.

Using these assumptions, equation (1) simplifies to

$$\hat{\Omega}\cdot\nabla\psi(\vec{r},\hat{\Omega})+\sigma_t(\vec{r})\psi(\vec{r},\hat{\Omega})=\sigma_s(\vec{r})\phi(\vec{r})+S_{EXT}(\vec{r}). \qquad (4)$$

The scalar flux, ϕ, is related to the angular flux, ψ, by

$$\phi(\vec{r})=\int d\Omega\,\psi(\vec{r},\hat{\Omega}) \qquad (5)$$

and the particle current, $\vec{J}$, is related to ψ by

$$\vec{J}(\vec{r}) = \int d\hat{\Omega}\, \Omega\, \psi(\vec{r},\hat{\Omega}). \qquad (6)$$

Angular flux by itself is not usually of interest. However when angular flux is integrated over angle, scalar flux and the vector current $\vec{J}$ can be produced. Scalar fluxes are needed to determine reaction rates such as fission and neutron activation rates. Vector currents are needed to determine leakage rates through boundaries or to track region to region particle movements.

Closed-form analytic solutions to equation (1) are known only in very limited cases. As a result, solutions are obtained by numerical means such as discrete ordinates or Monte Carlo methods.

2. Discrete Ordinates

The discrete ordinates method is a powerful but flexible numerical technique for obtaining global solutions to the neutral particle Boltzmann transport equation. This technique has evolved over several generations. Discrete ordinates focuses on separate treatment of angular and spatial variables, approximating each discretely by means of a numerical quadrature. This treatment provides great flexibility in the types of approximations used.

Angular dependence is approximated using a set of discrete directions (thus the name discrete ordinates). Scalar flux is obtained by performing a weighted sum on angular flux components for these discrete directions. Spatial dependence is approximated through spatial differencing or with a quadrature rule. The spatial quadrature effectively performs an inversion of the operator L of equation (1). A great deal of research has centered on finding spatial differencing techniques that provide improved speed, accuracy, and stability.

The discrete ordinates method has evolved over the past forty years, and has become one of the most widely used neutron transport methods available. It is simple to implement and allows for complexities such as anisotropic particle scatter and multiple energy group structure. Its independent treatment of space and angle provide flexibility to tailor quadratures and differencing to achieve desired speed and accuracy.

Use of discrete ordinates has many advantages but the technique also suffers from some numeric drawbacks. Discretizing energy, space, time, and position may require prohibitively large storage requirements. As a result, the ability to use coarsened spatial meshes is a desired trait. This dictates development of highly accurate spatial differencing schemes, as well as development of spatial quadratures that can take advantage of problem geometries.

Discretizing any variable inevitably produces errors in truncation. When the errors occur randomly, they reduce the accuracy of the method. However, when they are systematic in nature they can produce qualitatively incorrect results. Systematic errors that result from the discretization of angle are known as "ray effects." Another systematic error seen in discrete ordinates is known as "numerical diffusion." This type of error results from the inaccuracies associated with the spatial quadrature. Ray effects and numerical diffusion can be mitigated to some extent by refining the spatial and angular meshes.

B. Problem Statement

The purpose of this research is to derive, develop, implement, and evaluate the performance of a linear characteristic spatial quadrature on an arbitrary triangle. This effort will include choosing the appropriate coordinate frame, development of conservation relationships, and development of a triangle spatial mesh discrete ordinates algorithm. This algorithm will be compatible with the data structure required to handle triangles with arbitrary shape, orientation, and size, and with particles streaming in any direction.

C. Scope

The scope of this research is to develop and demonstrate the triangular linear characteristic spatial quadrature on a general triangle mesh. This linear characteristic quadrature will be valid for all triangle orientations and will match zeroth and first order integral moments exactly. The research will be restricted to two spatial dimensions. In addition, the following assumptions will be made:

1) flux-independent sources will be uniformly distributed by region and isotropic,

2) scattering will be isotropic in the lab frame,

3) the medium will be uniform, isotropic, and non-multiplying (no fission sources),

4) energy dependence will be simplified to one-group (monoenergetic), and

5) time dependence will be steady state.

Since arbitrary triangle meshes destroy the rectangle mesh discrete ordinates data structure (by not having rows and columns of cells), an alternative transport algorithm will be presented. A push-down stack process will replace the row/column cell-to-cell walk that is normally used. Although the inner structure of the general triangle discrete ordinates algorithm will change, the extensions of this algorithm to multi-group energy dependence, time dependence, and anisotropic scatter would be substantially the same as for its rectangular mesh counterpart.

D. Sequence of Presentation

Chapter II reviews previous work in two-dimensional Cartesian geometry. The Boltzmann transport equation is presented and discrete ordinates methods are discussed. A brief discussion of common angular quadratures is presented. In addition, common rectangular cell spatial quadratures are reviewed.

Chapter III presents development specific to the general triangle. Choice of coordinate frames is discussed. The triangle unit cell is defined and the case 0 triangle conservation relationships are presented. General triangle spatial quadratures are discussed. The general triangle step, step characteristic, and linear characteristic spatial quadratures are derived. Rotation and translation relationships for transformation of integral moments between coordinate frames are presented.

Chapter IV presents the algorithm of the TRISN program that was developed and implemented as a test bed for the triangle linear characteristic spatial quadrature. TRISN is compared and contrasted with SNXY, a previously validated two-dimensional discrete ordinates program.

Chapter V provides a series of test cases to demonstrate the linear characteristic spatial quadrature method. Benchmarks are presented showing general triangle linear characteristic spatial quadrature convergence rates using Lathrop's "square-in-a-square" problem. The asymptotic rate of convergence for triangle linear characteristic is compared to its rectangular spatial mesh analog as implemented in SNXY. A series of test cases is presented showing the advantages of general triangle meshes as opposed to orthogonal rectangle meshes. Results are presented and discussed. The advantages of general triangles over rectangles as the fundamental spatial mesh element is shown for non-square objects.

Chapter VI presents my conclusions and recommendations for further work.

II. Transport with Rectangular Spatial Cells

This chapter reviews the Boltzmann neutral particle transport equation particular to the simplified Cartesian geometry. It also reviews the rectangular cell conservation relationships and discrete ordinates angular quadratures. Finally, common spatial quadratures for the rectangle spatial cell are presented.

A. Discretizing in Angle for the Transport Equation

Simplifying the Boltzmann neutral particle transport equation to two-dimensional (x,y) geometry, isotropic scatter, and monoenergetic energy dependence produces the following

$$\left[\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}\right]\psi(x,y,\hat{\Omega})+\sigma_t(x,y)\,\psi(x,y,\hat{\Omega})=\sigma_s(x,y)\phi(x,y)+S_{EXT}(x,y). \quad (7)$$

If it is assumed that equation (7) applies to a finite set of angles $\hat{\Omega}_N$ in some angle quadrature set, then for the angle $\hat{\Omega}_n$, equation (7) can be written as

$$\left[\mu_n\frac{\partial}{\partial x}+\eta_n\frac{\partial}{\partial y}\right]\psi_n(x,y)+\sigma_t(x,y)\,\psi_n(x,y)=\sigma_s(x,y)\phi(x,y)+S_{EXT}(x,y), \quad (8)$$

where the scalar flux then relates to angular flux by a weighted sum over all $\hat{\Omega}_N$ and is given by

$$\phi(x,y)\approx\sum_{n=1}^{N}w_n\,\psi_n(x,y), \quad (9)$$

$$\sum_{n=1}^{N}w_n=1, \quad (10)$$

with (μ_n, η_n) and w_n being the set of direction cosines and associated weights.

B. Solution Techniques for the Boltzmann Equation

Two common approaches are used for obtaining numerical solutions to the neutral particle Boltzmann transport equation. These methods are Monte Carlo and discrete ordinates. The two methods are briefly discussed below.

1. Monte Carlo

Monte Carlo is a technique by which individual particle paths (histories) are simulated. These histories are accumulated and statistics are formed to measure desired processes. One advantage of Monte Carlo is the ability to handle very complex geometries. Another obvious advantage of the Monte Carlo technique is the continuous treatment of space and angle variables which eliminate the discretization errors associated with discrete ordinates. Monte Carlo errors take the form of stochastic uncertainties. One disadvantage of the technique is that the accuracy of the estimators is dependent on statistical variance, which may at times be expensive to compute [10:296]. Some variance reduction techniques (splitting, rouletting, absorption suppression, etc.) are available and improve the accuracy of estimators while reducing the computational expense. However, effective use of these techniques is as much an art as it is a science.

2. Discrete Ordinates

The method of discrete ordinates, sometimes called S_n, is substantially different from Monte Carlo. The discrete ordinates approach solves the Boltzmann transport equation by discretizing (meshing) the independent variables into phase elements. Since the angular flux ψ is implicit on both the left and right sides of the Boltzmann equation, discrete ordinates uses an iterative approach. One major advantage of the discrete ordinates method is that it provides global solutions for scalar fluxes (on the spatial mesh used). However, a disadvantage is the computational expense involved in performing the computations. For example, an n^2 increase in computational effort is expended for a factor of n refinement in a rectangular spatial mesh.

A typical discrete ordinates algorithm starts by establishing an initial guess for a scalar flux distribution on the spatial mesh. This guess might be zero throughout or some informed initial guess. The guess would be used to construct source information on each cell. Source information is used to construct the right-hand side of the Boltzmann equation. The source information coupled with boundary information would be used to perform particle transport from cell to cell in some sort of organized fashion in the direction of particle motion. Implicit in the cell-to-cell walk is some spatial quadrature which produces the angular flux for the cell of interest. The spatial quadrature effectively performs an inversion of the streaming and collision operator, L, of equation (1). Once the angular flux is calculated for every cell in the problem, a new angle would be chosen from the angle set and the walk would be repeated for all angles. An angular quadrature is performed on the angular flux components in each spatial cell to construct a scalar flux for each cell. The scalar flux is used to construct a new source distribution for the right-hand side of the Boltzmann transport equation and the next iteration. The iterative process is repeated until the entire problem converges to some predetermined limit.

Discrete ordinates can suffer from some computational problems. The discretization process invariably introduces error into the calculations. The errors mainly manifest themselves in two classes. These are errors associated with discretization in angle and errors associated with discretization in space.

Discrete representation of the continuous angular dependence results in truncation error. This type of error shows up as an inaccuracy in the scalar flux. This error is compounded because it is coupled through the scattering source to each iteration.

Ray effects are a particular type of systematic error associated with the angular quadrature that produce qualitatively incorrect results. These effects appear as unphysical spatial oscillations in the scalar flux. Ray effects are particularly pronounced in problems of small scattering-to-total cross-section ratio and in problems of small dimensions [10:195,6:255]. Ray effects can be mitigated by increasing the order of accuracy of the angular quadrature and by refining the angular mesh.

As with angle discretization, discretizing the space variable can produce truncation error. Systematic errors can also occur and are often hard to distinguish from systematic errors in angle. These errors occur mainly from inaccurate redistribution of flux from cell-to-cell as a result of approximations used in the spatial quadrature. These effects are sometimes referred to as numerical diffusion. Numerical diffusion can also be ıninimized by choosing spatial quadratures and mesh refinements that more accurately distribute the flux in the spatial cell.

C. Conservation in Two-Dimensional Cartesian Geometry

The conservation equations are a set of relationships that provide rules for relating the input and output information in a unit spatial cell. They relate integral moments of the angular flux to integral source moments. Conservation relationships provide necessary but not sufficient condˈitions to check the validity of a spatial quadrature. They also serve as a validity check for some complex spatial quadrature developments. For compactness of notation, the angular index, n, will be suppressed in the relationships that follow.

1. Cell-balance for the Rectangular Cell

The cell-balance equation is the lowest order of the conservation equations. It provides a relationship between cell-edge averages and cell-averages. A typical rectangle spatial cell can be seen in figure 1. Figure 1 shows the "characteristic line" which is oriented parallel to the projection of $\hat{\Omega}$ into the x-y plane. Cell-balance can be derived by integrating equation (7) across the spatial cell This produces

$$\frac{\psi_R-\psi_L}{\varepsilon_X}+\frac{\psi_T-\psi_B}{\varepsilon_Y}+\psi_A=\frac{S_A}{\sigma_t}, \tag{11}$$

where cell zeroth (average) moments ψ_A and S_A are defined as

$$\psi_A=\frac{1}{\Delta x\,\Delta y}\int_0^{\Delta y}dy\int_0^{\Delta x}dx\,\psi(x,y) \tag{12}$$

and

$$S_A = \frac{1}{\Delta x\, \Delta y} \int_0^{\Delta y} dy \int_0^{\Delta x} dx\, S(x,y). \tag{13}$$

ψ_L, ψ_R, ψ_B, and ψ_T are cell-edge zeroth (average) moments, defined as

$$\psi_B = \frac{1}{\Delta x} \int_0^{\Delta x} dx\, \psi(x,0), \tag{14}$$

$$\psi_T = \frac{1}{\Delta x} \int_0^{\Delta x} dx\, \psi(x,\Delta y), \tag{15}$$

$$\psi_L = \frac{1}{\Delta y} \int_0^{\Delta y} dy\, \psi(0,y), \tag{16}$$

and

$$\psi_R = \frac{1}{\Delta y} \int_0^{\Delta y} dy\, \psi(\Delta x,y), \tag{17}$$

while ε_X and ε_Y are optical thicknesses defined as

$$\varepsilon_X = \frac{\sigma_t\, \Delta x}{\mu} \tag{18}$$

and

$$\varepsilon_Y = \frac{\sigma_t\, \Delta y}{\eta}. \tag{19}$$

The cell zeroth moments are just averages for the appropriate distribution (flux or source) on the rectangle unit cell. Similarly, edge zeroth moments are the averages of the angular flux distribution along the appropriate edge. In this document, the terms "cell-average moment" and "cell zeroth moment" will be used interchangeably. In addition, the terms "cell edge average" and "cell edge zeroth moment" will also be used interchangeably.

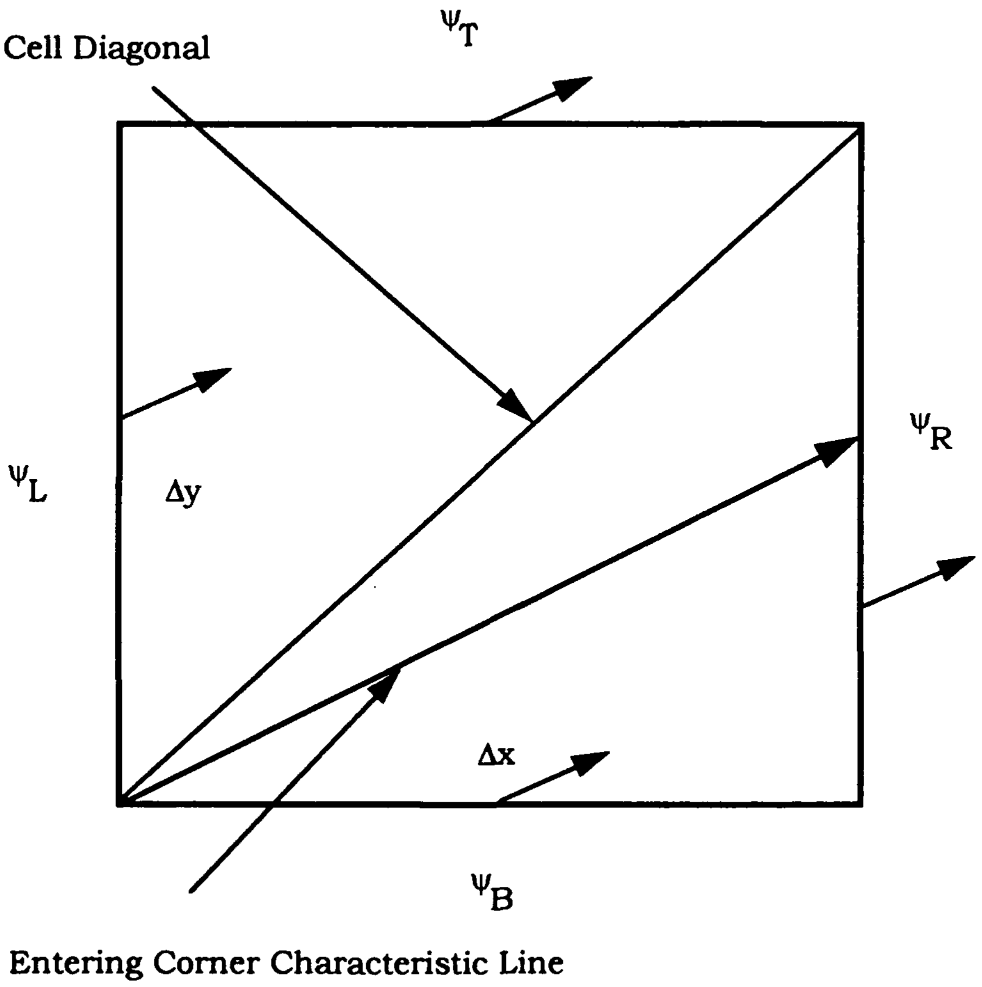


Figure 1. The Rectangle Unit Cell.

2. X-moment Conservation

Development of the x-moment conservation equation is very much like the cell-balance derivation. Equation (7) is multiplied by a linear weight and integrated across the rectangular spatial cell. The linear weight for the rectangle cell is usually taken as a multiple of the first order Legendre polynomial. The Legendre polynomials not only provide a basis for expansion of distributions, but exploit orthogonality on the rectangle spatial cell. Using $3P_1(x)$ as the weight, the x-moment conservation equation becomes

$$\frac{3(\psi_R+\psi_L-2\psi_A)}{\varepsilon_X}+\frac{(\theta_T-\theta_B)}{\varepsilon_Y}+\psi_X=\frac{S_X}{\sigma_t}, \tag{20}$$

where the cell x-moments are defined by

$$\psi_X=\frac{1}{\Delta x\,\Delta y}\int_0^{\Delta y}dy\int_0^{\Delta x}dx\;3P_1(x)\,\psi(x,y), \tag{21}$$

$$S_X=\frac{1}{\Delta x\,\Delta y}\int_0^{\Delta y}dy\int_0^{\Delta x}dx\;3P_1(x)S(x,y), \tag{22}$$

and the Legendre weight is shifted and scaled to give

$$P_1(x)=2\frac{x}{\Delta x}-1. \tag{23}$$

The first order edge-moments are defined by

$$\theta_B=\frac{1}{\Delta x}\int_0^{\Delta x}dx3P_1(x)\,\psi(x,0) \tag{24}$$

and

$$\theta_T=\frac{1}{\Delta x}\int_0^{\Delta x}dx\;3P_1(x)\,\psi(x,\Delta y). \tag{25}$$

3. Y-moment Conservation

The y-moment conservation equation is derived analogously to x-moment conservation, by multiplying equation (7) by the y-direction linear weight $3\,P_1(y)$ and integrating spatially across the cell domain. The y-moment equation is given by

$$\frac{3(\psi_T+\psi_B-2\,\psi_A)}{\varepsilon_Y}+\frac{(\theta_R-\theta_L)}{\varepsilon_X}+\psi_Y=\frac{S_Y}{\sigma_t}, \tag{26}$$

where the cell y-moments are defined as

$$\psi_Y=\frac{1}{\Delta x\,\Delta y}\int_0^{\Delta x}dx\int_0^{\Delta y}dy\;3\,P_1(y)\,\psi(x,y), \tag{27}$$

$$S_Y=\frac{1}{\Delta x\,\Delta y}\int_0^{\Delta x}dx\int_0^{\Delta y}dy\;3\,P_1(y)\,S(x,y), \tag{28}$$

and the Legendre weight is shifted and scaled to give

$$P_1(y)=2\frac{y}{\Delta y}-1. \tag{29}$$

D. Angular Quadratures

The angular quadrature provides a technique for discretizing the angle variable. It can be designed to take advantage of problem geometries, symmetries, and the level of accuracy desired.

1. Level Symmetric Quadratures

The level symmetric quadrature utilizes the same angle set for all three cardinal directions. This makes the quadrature invariant to 90 degree rotations. Level symmetric can be well suited to problems when a high degree of symmetry exists or when consistent angular differencing with respect to each axis is required [10:158]

2. Product Quadratures

Product quadratures are another class of angular quadrature. These quadratures are very flexible in their application. They can be designed to take advantage of symmetries in both polar and azimuthal angles. Several quadratures have been proposed such as the quadruple range quadrature and the Chebyshev-Gauss double-Gauss quadrature. Product quadratures can prove computationally cheap while providing a high degree of accuracy [1:299]. These quadratures can be tailored to the degree of accuracy required for a particular problem and they can also be designed to take advantage of unusual problem symmetries.

E. Spatial Quadratures for Rectangular Cells

This is a review of some common two-dimensional Cartesian spatial quadratures that use a rectangle spatial cell. Figure 1 illustrates a typical spatial mesh cell. Without loss of generality, figure 1 shows an angle in the first quadrant where particles flow from the bottom left to the upper right. This implies that particles enter at the bottom and left edges and exit the top and right edges. As a result, the bottom and left cell edges contain input information and top and right edges require information from the spatial quadrature for output. Any other angle orientations can be handled through reflections of the cell about the cardinal axes. In addition, Δx and Δy do not need to be equal. Order of convergence discussions that follow assume that if Δx and Δy are not equal, then Δx is the greater of the two.

The relationships presented in this section use the notation introduced by Walters [16:193-6] and later expanded by Mathews [12:419-457,13]. Step, step characteristic, and linear characteristic are presented in detail to allow comparison with their triangle analogs presented in the next chapter. For completeness, diamond difference relationships are also presented. Some other higher order spatial quadrature methods are discussed briefly. The formulae for these methods are omitted due to their complexity.

1. Rectangular Step Method

The step method is the lowest order and simplest to implement of the spatial quadrature methods. It assumes that the angular flux is constant in the interior of the spatial cell. The step method also assumes that the output edge averages are equal to the cell-average flux. With these assumptions, the rectangle cell-balance equation reduces to one unknown. Solving the simplified cell-balance relationship for cell-average flux within the cell produces the following

$$\psi_A = \frac{\left(S_A + \psi_L \frac{\eta}{\Delta y} + \psi_B \frac{\mu}{\Delta x}\right)}{\left(\sigma_t + \frac{\eta}{\Delta y} + \frac{\mu}{\Delta x}\right)} \tag{30}$$

with

$$\psi_R = \psi_A \tag{31}$$

and

$$\psi_T = \psi_A. \tag{32}$$

The rectangle step method is a positive method (producing positive results for positive inputs). However, its low order accuracy limits its utility.

2. Rectangle Diamond Difference

The diamond difference approximation is one of the most commonly used spatial quadrature methods. It assumes that the angular flux within a spatial cell varies linearly across the cell in both x and y and is continuous at cell boundaries. As a result, the cell-average angular flux can be approximated by an average of the edge-fluxes in either direction. This produces two auxiliary relationships:

$$\psi_A = \frac{(\psi_R + \psi_L)}{2} \tag{33}$$

and

$$\psi_A = \frac{(\psi_T + \psi_B)}{2}. \tag{34}$$

These relationships can be combined with the cell-balance relationship of equation (11) to give the following solutions

$$\psi_A = \frac{\left(S_A + 2\left(\psi_L \frac{\mu}{\Delta x} + \psi_B \frac{\eta}{\Delta y}\right)\right)}{\left(\sigma_t + 2\left(\frac{\mu}{\Delta x} + \frac{\eta}{\Delta y}\right)\right)}, \tag{35}$$

$$\psi_R = 2\psi_A - \psi_L, \tag{36}$$

and

$$\psi_T = 2\psi_A - \psi_B. \tag{37}$$

Diamond difference provides $O(\Delta x^2)$ convergence [6:78] and positivity for optically thin spatial meshes. However, if the spatial mesh becomes too coarse negative fluxes result. These negative fluxes can affect accuracy [8]. Negative flux fix-ups can be implemented to remedy the negativity problems, but can further degrade accuracy.

3. Rectangle Step Characteristic

The rectangle step characteristic method for two-dimensional Cartesian geometry was developed by Lathrop [8]. This method assumes the source distribution within the cell and the cell edge angular flux can be represented by constants.

The rectangle step characteristic method is analytic throughout the unit cell except along the entering corner characteristic line which marks a change in input boundary conditions. The piecewise analytic behavior in the cell allows piecewise integration across the cell. Using the integral form of equation (7) and applying moments definitions discussed above, the output moments can be calculated by direct analytic integration and are given by

$$\psi_A = \begin{bmatrix} (1-\varepsilon_R)\psi_L \mathcal{M}_0(\varepsilon_X) + \\ (\varepsilon_R(\psi_L + \psi_B) + (1-\varepsilon_R)q_A)\mathcal{M}_1(\varepsilon_X) + \\ \varepsilon_R q_A \mathcal{M}_2(\varepsilon_X) \end{bmatrix}, \tag{38}$$

$$\psi_R = \begin{bmatrix} (1-\varepsilon_R)\psi_L e^{-\varepsilon_X} + \\ (\varepsilon_R \psi_B + (1-\varepsilon_R)q_A)\mathcal{M}_0(\varepsilon_X) + \\ \varepsilon_R q_A \mathcal{M}_1(\varepsilon_X) \end{bmatrix}, \tag{39}$$

and

$$\psi_T = \psi_L \mathcal{M}_0(\varepsilon_X) + q_A \mathcal{M}_1(\varepsilon_X), \tag{40}$$

where

$$\varepsilon_R = \frac{\varepsilon_X}{\varepsilon_Y} = \frac{\Delta x}{\mu} \frac{\eta}{\Delta y} \tag{41}$$

and

$$q_A = S_A \frac{\Delta x}{\mu}. \tag{42}$$

In addition, exponential moment functions, $\mathcal{M}_n$, are defined as

$$\mathcal{M}_n(x) = \int_0^1 (1-t)^n e^{-xt}\, dt. \tag{43}$$

The exponential moment functions are a generalization of the recursively defined functions introduced by Walters [16:193] which are given by

$$\mathcal{M}_0(x) = \frac{\left(1-e^{-x}\right)}{x} \tag{44}$$

and

$$\mathcal{M}_n(x) = \frac{\left(1-n\,\mathcal{M}_{n-1}(x)\right)}{x}. \tag{45}$$

(Walters used the notation $P_i(x)$ which is readily confused with Legendre polynomial notation. Mathews [12:431] changed this notation to $p_i(x)$ in an attempt to avoid this confusion.) The exponential moment functions provide a means for removing the removable singularities that appear during characteristic type spatial quadrature derivations. The moment formulae above apply only to angles in the principle octant that project into the (x,y) plane at angles below the cell diagonal shown in figure 1. All other relationships can be obtained by reflection or symmetry arguments.

The step characteristic spatial quadrature obeys zeroth order conservation relationships and, like diamond difference, provides $O\left(\Delta x^2\right)$ convergence [6:78]. One great advantage of step characteristic is its positivity. Unlike diamond difference, there is no need for any type of flux fix-up.

4. Rectangle Linear Characteristic

The rectangle linear characteristic spatial quadrature was developed by Larsen [5]. Linear characteristic is similar in development to step characteristic. The linear characteristic method provides an improved approximation to the source distribution. It assumes the source can be represented by an inclined plate. The source distribution is expanded in Legendre polynomials and given by

$$S(x,y) = S_A\,P_0(x)P_0(y) + S_X\,P_1(x)P_0(y) + S_Y\,P_0(x)P_1(y). \tag{46}$$

Using the integral form of equation (7) and applying moments definitions discussed above, the output moments can be calculated by direct analytic integration. The moments ψ_T and ψ_R are given by

$$\psi_T = \begin{bmatrix} \psi_L\,\mathcal{M}_0(\varepsilon_X) + \theta_L\left(\mathcal{M}_0(\varepsilon_X) + 2\,\varepsilon_R\left(\mathcal{M}_1(\varepsilon_X) - \mathcal{M}_0(\varepsilon_X)\right)\right) + q_A\,\mathcal{M}_1(\varepsilon_X) + \\ q_X\left(\mathcal{M}_2(\varepsilon_X) - \mathcal{M}_1(\varepsilon_X)\right) + q_Y\left(\mathcal{M}_1(\varepsilon_X) + 2\,\varepsilon_R\left(\mathcal{M}_2(\varepsilon_X) - \mathcal{M}_1(\varepsilon_X)\right)\right) \end{bmatrix} \tag{47}$$

and

$$\psi_R = \begin{bmatrix} (1-\varepsilon_R)e^{-\varepsilon_X}\psi_L - (1-\varepsilon_R)\varepsilon_R\,e^{-\varepsilon_X}\theta_L + \varepsilon_R\,\mathcal{M}_0(\varepsilon_X)\,\psi_B + \\ \varepsilon_R\,\theta_B\left(2\,\mathcal{M}_1(\varepsilon_X) - \mathcal{M}_0(\varepsilon_X)\right) + q_A\left(\mathcal{M}_0(\varepsilon_X)(1-\varepsilon_R) + \varepsilon_R\,\mathcal{M}_1(\varepsilon_X)\right) + \\ q_X\left(\mathcal{M}_0(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_1(\varepsilon_X)(2 - 3\,\varepsilon_R) + 2\,\varepsilon_R\,\mathcal{M}_2(\varepsilon_X)\right) + \\ q_Y\left(\mathcal{M}_0(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_1(\varepsilon_X)(1 - 2\,\varepsilon_R) + \varepsilon_R\,\mathcal{M}_2(\varepsilon_X)\right) \end{bmatrix}. \tag{48}$$

The edge first moments are given by

$$\theta_T = \begin{bmatrix} 3\,\psi_L\left(\mathcal{M}_0(\varepsilon_X)-2\mathcal{M}_1(\varepsilon_X)\right)+3\,q_A\left(\mathcal{M}_1(\varepsilon_X)-\mathcal{M}_2(\varepsilon_X)\right)+ \\ 3\,\theta_L\left(\mathcal{M}_0(\varepsilon_X)(1-2\,\varepsilon_R)-2\,\mathcal{M}_1(\varepsilon_X)(1-3\,\varepsilon_R)-4\,\varepsilon_R\,\mathcal{M}_2(\varepsilon_X)\right)+ \\ q_X\left(-3\,\mathcal{M}_1(\varepsilon_X)+6\,\mathcal{M}_2(\varepsilon_X)-2\,\mathcal{M}_3(\varepsilon_X)\right)+ \\ 3\,q_Y\left(\mathcal{M}_1(\varepsilon_X)(1-2\,\varepsilon_R)+\mathcal{M}_2(\varepsilon_X)(4\,\varepsilon_R-1)+2\,\varepsilon_R\,\mathcal{M}_3(\varepsilon_X)\right) \end{bmatrix} \tag{49}$$

and

$$\boldsymbol{\theta}_R = \boldsymbol{\theta}_R^{\Psi_L} + \boldsymbol{\theta}_R^{\Psi_B} + \boldsymbol{\theta}_R^{\theta_L} + \boldsymbol{\theta}_R^{\theta_B} + \boldsymbol{\theta}_R^{S_A} + \boldsymbol{\theta}_R^{S_X} + \boldsymbol{\theta}_R^{S_Y} \tag{50}$$

where

$$\theta_R^{\psi_L} = 3\,(\varepsilon_R-1)\left(\varepsilon_X\,\mathcal{M}_0(\varepsilon_X)-1\right)\varepsilon_R\,\psi_L \tag{51}$$

$$\boldsymbol{\theta}_R^{\theta_L} = \left(1-3\boldsymbol{\varepsilon}_R+2\boldsymbol{\varepsilon}_R{}^3\right)e^{-\varepsilon_X}\,\boldsymbol{\theta}_L \tag{52}$$

$$\theta_R^{\psi_B} = 3\,\varepsilon_R\,\psi_B\left(\mathcal{M}_0(\varepsilon_X)(2\,\varepsilon_R-1)-2\,\varepsilon_R\,\mathcal{M}_1(\varepsilon_X)\right) \tag{53}$$

$$\theta_R^{\theta_B} = 3\,\varepsilon_R\,\theta_B\left(\mathcal{M}_0(\varepsilon_X)(1-2\,\varepsilon_R)+2\,\mathcal{M}_1(\varepsilon_X)\,(3\,\varepsilon_R-1)-4\,\mathcal{M}_2(\varepsilon_X)\right) \tag{54}$$

$$\theta_R^{S_A} = 3\,q_A\,\varepsilon_R\left(\mathcal{M}_0(\varepsilon_X)(1-\ \varepsilon_R)+\mathcal{M}_1(\varepsilon_X)\,(2\,\varepsilon_R-1)-\varepsilon_R\,\mathcal{M}_2(\varepsilon_X)\right) \tag{55}$$

$$\begin{aligned} \theta_R^{S_X} = 3\,q_X\,\varepsilon_R\,(\mathcal{M}_0(\varepsilon_X)(\varepsilon_R-1)+\mathcal{M}_1(\varepsilon_X)\,(3-4\ \varepsilon_R) \\ +\mathcal{M}_2(\varepsilon_X)(5\,\varepsilon_R-2)-2\,\varepsilon_R\,\mathcal{M}_3(\varepsilon_X)) \end{aligned} \tag{56}$$

$$\begin{aligned} \boldsymbol{\theta}_R^{S_Y} = q_Y\,(\mathcal{M}_0(\boldsymbol{\varepsilon}_X)\,\left(1-3\ \boldsymbol{\varepsilon}_R+2\boldsymbol{\varepsilon}_R{}^3\right)+3\boldsymbol{\varepsilon}_R\,\mathcal{M}_1(\boldsymbol{\varepsilon}_X)\left(1-\boldsymbol{\varepsilon}_R{}^2\right)+ \\ 6\boldsymbol{\varepsilon}_R{}^3\,\mathcal{M}_2(\boldsymbol{\varepsilon}_X)-2\boldsymbol{\varepsilon}_R{}^3\,\mathcal{M}_3(\boldsymbol{\varepsilon}_X). \end{aligned} \tag{57}$$

The notation used here is that $X_i^{Y_j}$ is the contribution to output moment X_i from input moment Y_j.

The cell-average angular flux ψ_A is given by

$$\psi_A = \psi_A^{\Psi_L} + \psi_A^{\Psi_B} + \psi_A^{\theta_L} + \psi_A^{\theta_B} + \psi_A^{S_A} + \psi_A^{S_X} + \psi_A^{S_Y} \tag{58}$$

where

$$\psi_A^{\psi_L} = \psi_L \left(\mathcal{M}_0(\varepsilon_X)(1-\varepsilon_R) + \varepsilon_R \mathcal{M}_1(\varepsilon_X) \right) \tag{59}$$

$$\psi_A^{\theta_L} = \varepsilon_R \, \theta_L \left(\mathcal{M}_0(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_1(\varepsilon_X)(1 - 2\,\varepsilon_R) + \varepsilon_R \mathcal{M}_2(\varepsilon_X) \right) \tag{60}$$

$$\psi_A^{\psi_B} = \varepsilon_R \, \psi_B \, \mathcal{M}_1(\varepsilon_X) \tag{61}$$

$$\psi_A^{\theta_B} = \varepsilon_R \, \theta_B \left(\mathcal{M}_2(\varepsilon_X) - \mathcal{M}_1(\varepsilon_X) \right) \tag{62}$$

$$\psi_A^{S_A} = q_A \left(\mathcal{M}_1(\varepsilon_X)(1 - \varepsilon_R) + \varepsilon_R \mathcal{M}_2(\varepsilon_X) \right) \tag{63}$$

$$\psi_A^{S_X} = q_X \left(\mathcal{M}_1(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_2(\varepsilon_X)(1 - 2\,\varepsilon_R) + \varepsilon_R \mathcal{M}_3(\varepsilon_X) \right) \tag{64}$$

$$\psi_A^{S_Y} = q_Y \, \varepsilon_R \left(\mathcal{M}_1(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_2(\varepsilon_X)(1 - 2\,\varepsilon_R) + \varepsilon_R \mathcal{M}_3(\varepsilon_X) \right). \tag{65}$$

The cell x-moment is given by

$$\boldsymbol{\psi}_X = \boldsymbol{\psi}_X^{\psi_L} + \boldsymbol{\psi}_X^{\psi_B} + \boldsymbol{\psi}_X^{\theta_L} + \boldsymbol{\psi}_X^{\theta_B} + \boldsymbol{\psi}_X^{S_A} + \boldsymbol{\psi}_X^{S_X} + \boldsymbol{\psi}_X^{S_Y}, \tag{66}$$

where

$$\psi_X^{\psi_L} = 3\, \psi_L \left(\mathcal{M}_0(\varepsilon_X)(1 - \varepsilon_R) + \mathcal{M}_1(\varepsilon_X)(3\ \varepsilon_R - 2) - 2\, \varepsilon_R \mathcal{M}_2(\varepsilon_X) \right) \tag{67}$$

$$\begin{aligned} \psi_X^{\theta_L} = 3\, \varepsilon_R \, \theta_L \, (\mathcal{M}_0(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_1(\varepsilon_X)(3 - 4\, \varepsilon_R) + \\ \mathcal{M}_2(\varepsilon_X)(5\, \varepsilon_R - 2) - 2\, \varepsilon_R \mathcal{M}_3(\varepsilon_X)) \end{aligned} \tag{68}$$

$$\psi_X^{\psi_B} = 3\, \varepsilon_R \, \psi_B \left(\mathcal{M}_1(\varepsilon_X) - \mathcal{M}_2(\varepsilon_X) \right) \tag{69}$$

$$\psi_X^{\theta_B} = \varepsilon_R \, \theta_B \left(-3\, \mathcal{M}_1(\varepsilon_X) + 6\, \mathcal{M}_2(\varepsilon_X) - 2\, \mathcal{M}_3(\varepsilon_X) \right) \tag{70}$$

$$\psi_X^{S_A} = 3\ q_A \left(\mathcal{M}_1(\varepsilon_X)(1 - \varepsilon_R) + \mathcal{M}_2(\varepsilon_X)(2\ \varepsilon_R - 1) - \ \varepsilon_R \mathcal{M}_3(\varepsilon_X) \right) \tag{71}$$

$$\begin{aligned} \psi_X^{S_X} = q_X \, (3\, \mathcal{M}_1(\varepsilon_X)(\varepsilon_R - 1) + 3 \mathcal{M}_2(\varepsilon_X)(2 - 3\, \varepsilon_R) + \\ 2\, \mathcal{M}_3(\varepsilon_X)(4\ \varepsilon_R - 1) - 2\ \varepsilon_R \mathcal{M}_4(\varepsilon_X)) \end{aligned} \tag{72}$$

$$\psi_Y^{S_X} = 3\,\varepsilon_R\, q_Y\, (\mathcal{M}_1(\varepsilon_X)(\varepsilon_R - 1) + \mathcal{M}_2(\varepsilon_X)(2 - 3\,\varepsilon_R) + \\ \mathcal{M}_3(\varepsilon_X)(3\,\varepsilon_R - 1) - \varepsilon_R\, \mathcal{M}_4(\varepsilon_X)). \tag{73}$$

The cell y-moment is given by

$$\psi_Y = \psi_Y^{\psi_L} + \psi_Y^{\psi_B} + \psi_Y^{\theta_L} + \psi_Y^{\theta_B} + \psi_Y^{S_A} + \psi_Y^{S_X} + \psi_Y^{S_Y} \tag{74}$$

where

$$\psi_Y^{\psi_L} = 3\,\varepsilon_R\, \psi_L \left(\mathcal{M}_0(\varepsilon_X)(1 - \varepsilon_R) + \mathcal{M}_1(\varepsilon_X)(2\,\varepsilon_R - 1) - \varepsilon_R\, \mathcal{M}_2(\varepsilon_X)\right) \tag{75}$$

$$\psi_Y^{\theta_L} = \theta_L\, (\mathcal{M}_0(\varepsilon_X)\left(1 - 3\,\varepsilon_R + 2\,\varepsilon_R^3\right) + 3\,\varepsilon_R\, \mathcal{M}_1(\varepsilon_X)\left(1 - 2\,\varepsilon_R^2\right) + \\ 6\,\varepsilon_R^3\, \mathcal{M}_2(\varepsilon_X) - 2\,\varepsilon_R^3\, \mathcal{M}_3(\varepsilon_X)) \tag{76}$$

$$\psi_Y^{\psi_B} = 3\,\varepsilon_R\, \psi_B \left(\mathcal{M}_1(\varepsilon_X)(2\,\varepsilon_R - 1) - 2\,\varepsilon_R\, \mathcal{M}_2(\varepsilon_X)\right) \tag{77}$$

$$\psi_Y^{\theta_B} = 3\,\varepsilon_R\, \theta_B \left(\mathcal{M}_1(\varepsilon_X)(1 - 2\,\varepsilon_R) + \mathcal{M}_2(\varepsilon_X)(4\,\varepsilon_R - 1) - 2\,\varepsilon_R\, \mathcal{M}_3(\varepsilon_X)\right) \tag{78}$$

$$\psi_Y^{S_A} = 3\,\varepsilon_R\, q_A \left(\mathcal{M}_1(\varepsilon_X)(1 - \varepsilon_R) + \mathcal{M}_2(\varepsilon_X)(2\,\varepsilon_R - 1) - \varepsilon_R\, \mathcal{M}_3(\varepsilon_X)\right) \tag{79}$$

$$\psi_Y^{S_X} = q_X \left(\mathcal{M}_2(\varepsilon_X) - \mathcal{M}_1(\varepsilon_X)\right) \tag{80}$$

$$\psi_Y^{S_Y} = q_X\, (\mathcal{M}_1(\varepsilon_X)\left(1 - 3\,\varepsilon_R + 2\,\varepsilon_R^3\right) + 3\,\varepsilon_R\, \mathcal{M}_2(\varepsilon_X)\left(1 - 2\,\varepsilon_R^2\right) + \\ 6\,\varepsilon_R^3\, \mathcal{M}_3(\varepsilon_X) - 2\,\varepsilon_R^3\, \mathcal{M}_4(\varepsilon_X)). \tag{81}$$

Analogous to q_A, q_X and q_Y are defined as

$$q_X = S_X \frac{\Delta x}{\mu} \tag{82}$$

and

$$q_Y = S_Y \frac{\Delta x}{\mu} \tag{83}$$

(Note: q_Y should not be confused with $S_Y \frac{\Delta y}{\eta}$).

As with the step method, the linear characteristic moment formulae apply only to angles in the principle octant that project into the (x, y) plane at angles below the cell diagonal shown in figure 1. All other relationships can again be obtained by reflection or symmetry arguments.

Linear characteristic obeys zeroth and first moment conservation relationships and tends to be more accurate than step characteristic on similar spatial meshes. The linear characteristic method can be shown analytically to converge as $O\left(\Delta x^3\right)$, but under certain circumstances it has been observed to super converge at rates as high as $O\left(\Delta x^4\right)$ [6:80]. However unlike step characteristic, linear characteristic is not a strictly positive method. As a result, fix-ups may be needed with coarse spatial meshes.

5. Other Rectangle Quadratures

Other rectangle spatial quadratures have been developed and implemented. A class of methods known as the nodal methods [2,16,18] prescribe analytic integration in the spatial variable. The most common of the nodal methods is linear nodal. Linear nodal is a spatial quadrature used in some production codes but can suffer from negative flux problems on coarse spatial meshes.

Another class of methods is the adaptive methods [12,13]. These methods are a variation of the characteristic methods. Adaptive methods use adaptive, piecewise-defined source representations to remedy the need for negative flux fix-ups. These methods allow optically thick spatial cells while maintaining high accuracy. The adaptive methods are particularly well suited to the deep penetration problem.

III. General Triangle Spatial Quadrature Development

This chapter presents the derivation of a new discrete ordinates spatial quadrature for solution of the Boltzmann neutral particle transport equation on an arbitrarily oriented and shaped triangle. The method is a linear characteristic method using arbitrarily oriented triangles.

A. The Local Coordinate System

Allowing for an arbitrarily oriented triangle makes the necessary mathematical manipulations prohibitively complex. A fortuitous orientation and placement of coordinate axes can greatly simplify the necessary manipulations. Rectangle spatial quadrature development focuses on a unit rectangle spatial cell similar to the one depicted in figure 1. A standardized unit cell greatly simplifies quadrature development and allows more complicated approximations. A triangle unit cell is desired. Many different positions and orientations of the local frame were attempted. The local system chosen is the case 0 system discussed below. This case 0 system is not the only system that could be used, but it is most effective in simplifying the characteristic integration and the integral moment splitting and assembly (discussed later).

Consider the arbitrarily oriented triangle in two-dimensional (x,y) Cartesian geometry of figure 2. Figure 2 shows that particle streaming through an arbitrarily oriented triangle can be separated into two cases dependent on triangle orientation, shape, and direction of particle streaming. Input information (particle entering boundary) is located along the edge(s) where particles enter the cell. Output information is located on the edge(s) where particles leave the cell.

A degenerate case occurs when particles stream in a direction such that their travel is exactly parallel to one edge. This would mean one triangle edge would contain input information, one edge would contain output information, and one edge would contain neither (no current crosses this edge). This degenerate case will be referred to as the case 0 orientation. The case 0 orientation is particularly interesting because it leads to tractable derivations and formulae.

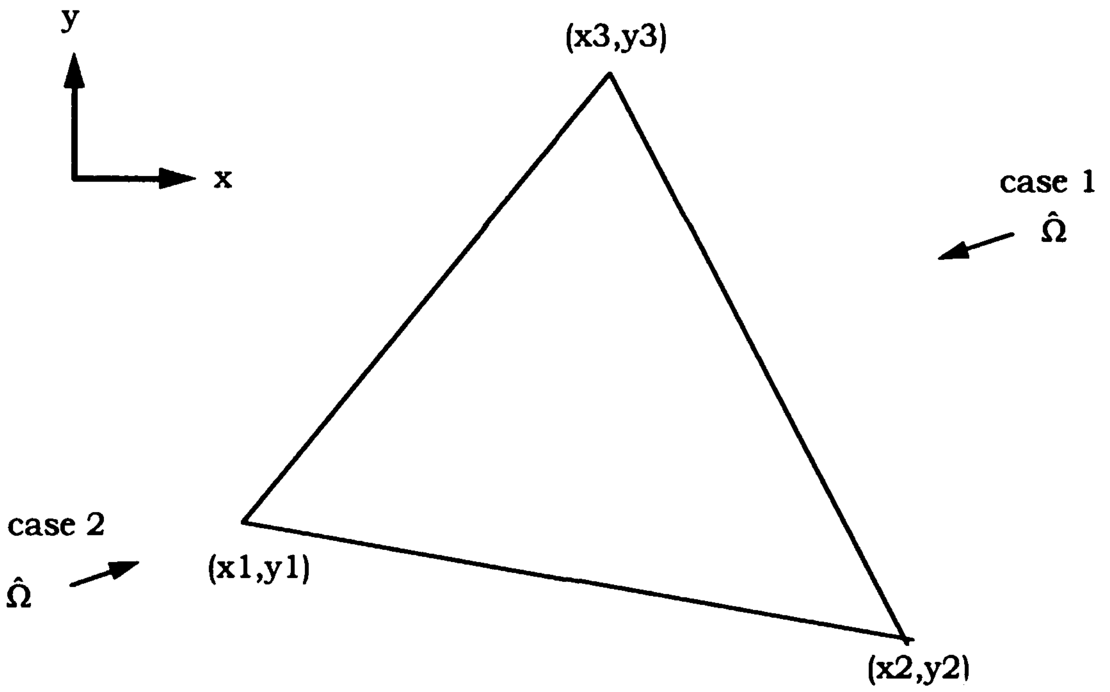


Figure 2. An Arbitrary 'riangle.

Figure 2 shows particles may enter in a direction such that one edge has input information and two edges have output information. This case will be referred to as the case 1 (one input) orientation. Particles may enter in a direction such that two edges are inputs and one edge is an output. This case is referred to as the case 2 (two input) orientation and is also shown in figure 2. These three cases exhaust all the possibilities for streaming across a triangle of any orientation and any streaming direction. The goal is to choose a coordinate system that takes advantage of the properties of the case 0 triangle.

Local u and v axes can be constructed such that u is oriented parallel to and in the same direction as particles are traveling and v is perpendicular to u and maintains a right-hand sense. An origin for the local (u,v) coordinate system can be assigned to the triangle boundary using the following rules:

1) If the triangle has a case 0 orientation, the (u,v) origin is placed at the intersection of the edge parallel to streaming and the input edge (see figure 3).

2) If the triangle has a case 1 orientation, the (u,v) origin is placed at the intersection of the input edge and the particle streaming line through the intersection of the output edges (see figure 4).

3) If the triangle has a case 2 orientation, the (u,v) origin is placed at the intersection of the input edges (see figure 5).

This choice of coordinates for the local frame accomplishes several goals. First, case 1 and case 2 orientations degenerate into the sum of two case 0 oriented triangles (see figures 4 and 5). This implies the spatial quadratures need only be developed for case 0 orientations. Next, case 1 and case 2 oriented triangles subdivide into case 0 oriented sub triangles that share a common origin and axes which make integral moment splitting and assembly very simple. In the local (u,v) frame, transport is reduced to one-dimension and only boundary conditions are two-dimensional. As a result, the Boltzmann equation takes on a simpler form and makes the mathematical relationships much simpler.

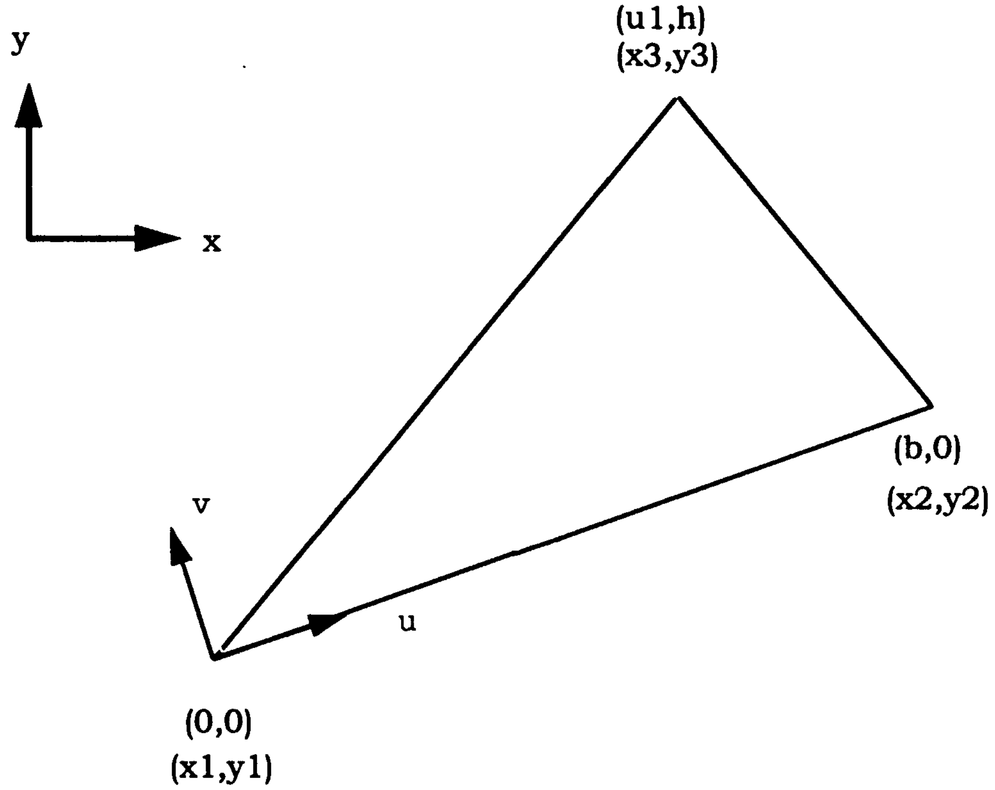


Figure 3. Case 0 Triangle.

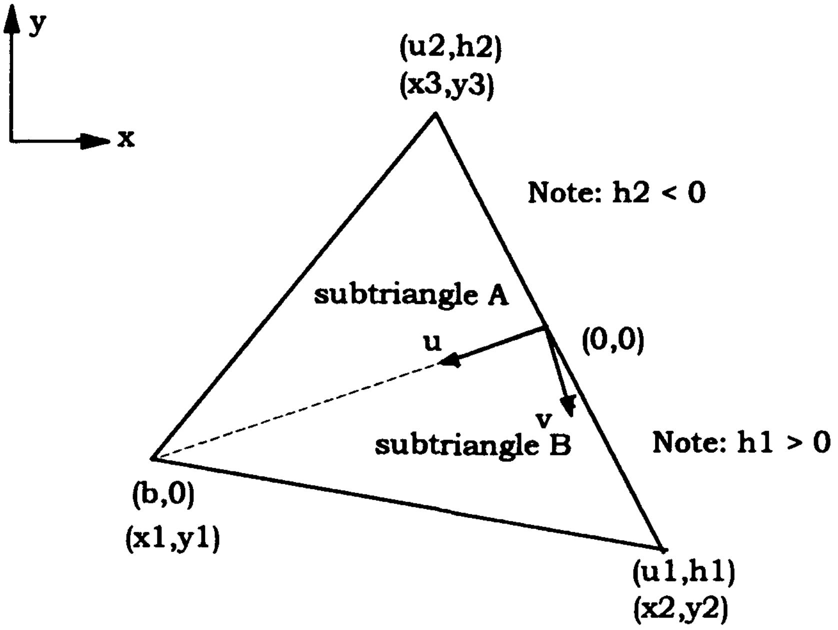


Figure 4. Case 1 Triangle.

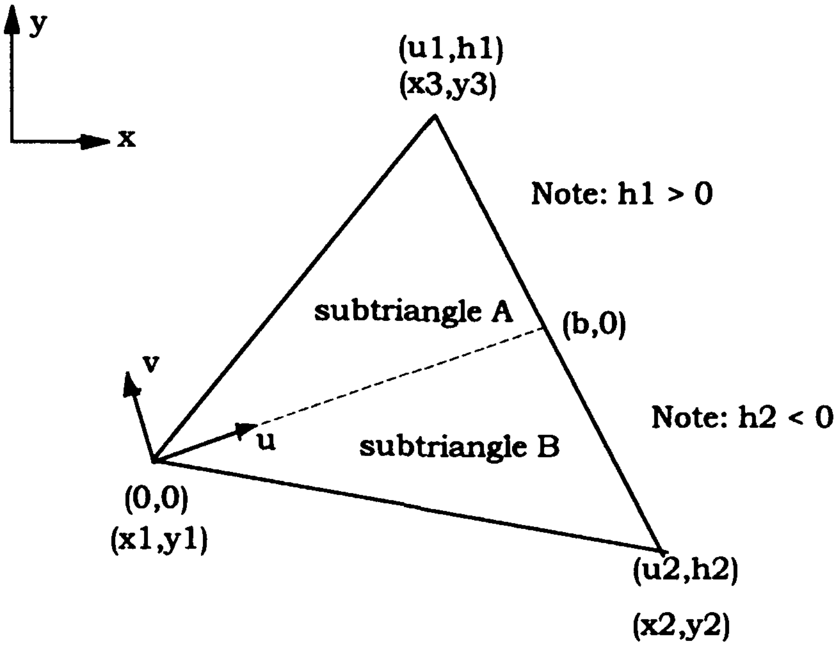


Figure 5. Case 2 Triangle.

Construction of a coordinate frame in which particle travel is constrained to one direction (along u) changes the form of the Boltzmann transport equation. The change to the local (u,v) frame simplifies representation of the Boltzmann form of equation (4) to the following

$$\mu' \frac{\partial\psi(u,v,\mu')}{\partial u} + \sigma_t(u,v)\,\psi(u,v,\mu') = \sigma_s(u,v)\phi(u,v) + S_{EXT}(u,v), \tag{84}$$

where μ' is the u direction cosine of $\hat{\Omega}$ and is given by

$$\mu' = \sqrt{\mu^2 + \eta^2} = \sqrt{1 - \xi^2}, \tag{85}$$

and μ, η, and ξ are the x, y, and z direction cosines of $\hat{\Omega}$, respectively and are given by

$$\mu = \hat{\Omega}_x = \hat{\Omega}\cdot\hat{e}_x, \tag{86}$$

$$\eta = \hat{\Omega}_y = \hat{\Omega}\cdot\hat{e}_y, \tag{87}$$

and

$$\xi = \hat{\Omega}_z = \hat{\Omega}\cdot\hat{e}_z. \tag{88}$$

Analytic integration of equation (84) along streaming direction u produces another form of the Boltzmann neutral particle transport equation known as the integral form. In the local frame this equation is given by

$$\psi(u,v) = \psi(u_L(v),v)\exp\left[\frac{-\sigma_t(u - u_L(v))}{\mu'}\right] + \int_{u_L(v)}^{u} \frac{du'}{\mu'} S(u',v)\exp\left[\frac{-\sigma_t(u - u')}{\mu'}\right] \tag{89}$$

where $u_L(v)$ is the u-coordinate at which particles enter the triangle input edge, $\psi(u_L(v),v)$ is the angular flux value at that location and $S(u,v)$ is given by

$$S(u,v)=\sigma_S(u,v)\phi(u,v)+S_{EXT}(u,v). \qquad (90)$$

For the case 0 triangle $u_L(v)$, is given by

$$u_L(v)=u_1\frac{v}{h} \qquad (91)$$

B. Conservation Equations

Conservation equations are a necessary but not sufficient condition needed to derive the spatial quadratures discussed later. They provide a validity check for derivation of the spatial quadratures. These equations can be obtained by taking integral moments of the Boltzmann neutral particle transport equation.

1. Case 0 Cell-Balance

For the case 0 orientation, the zeroth order or cell-balance equation can be derived by taking the neutral particle Boltzmann transport equation, equation (84), and integrating across the domain of the triangle as follows

$$\int_0^h dv \int_{u_L(v)}^{u_R(v)} du\left[\mu'\frac{d\psi(u,v,\mu')}{du}+\sigma_t(u,v)\psi(u,v,\mu')=\sigma_s(u,v)\phi(u,v)+S_{EXT}(u,v)\right]. \qquad (92)$$

Assuming the area of the triangle is sufficiently small that cross-sections can be approximated as constants, equation (92) simplifies to

$$\sigma_t\,\psi_A+\frac{2}{b}\mu'(\psi_{OUT}-\psi_{IN})=S_A \qquad (93)$$

where

$$\sigma_s(u,v) \approx \sigma_s, \tag{94}$$

and

$$\sigma_t(u,v) \approx \sigma_t. \tag{95}$$

The triangle zeroth (average) moments for source and angular flux are given by

$$\psi_A = \frac{1}{A}\int_\Delta dA\, \psi(\vec{r}) \tag{96}$$

and

$$S_A = \frac{1}{A}\int_\Delta dA\; S(\vec{r}) \tag{97}$$

where

$$S(\vec{r}) = \sigma_s(\vec{r})\phi(\vec{r}) + S_{EXT}(\vec{r}). \tag{98}$$

ψ_{IN} and ψ_{OUT} are angular flux averages along the input and output edges, respectively, and are defined by

$$\psi_{IN} = \frac{1}{L_{IN}}\int ds_{IN}\; \psi(s_{IN}) \tag{99}$$

and

$$\psi_{OUT} = \frac{1}{L_{OUT}}\int ds_{OUT}\; \psi(s_{OUT}) \tag{100}$$

where L_{IN} , L_{OUT}, A, and b are the lengths of the input and output triangle edges, the triangle area, and the triangle base. The variables s_{IN} and s_{OUT} are contour integration variables which range from zero to the length of the corresponding triangle edge and whose origins are defined in counter clockwise fashion, as shown in figure 6. Equation (93) is referred to as the case 0 cell-balance equation and is valid only on the case 0 triangle.

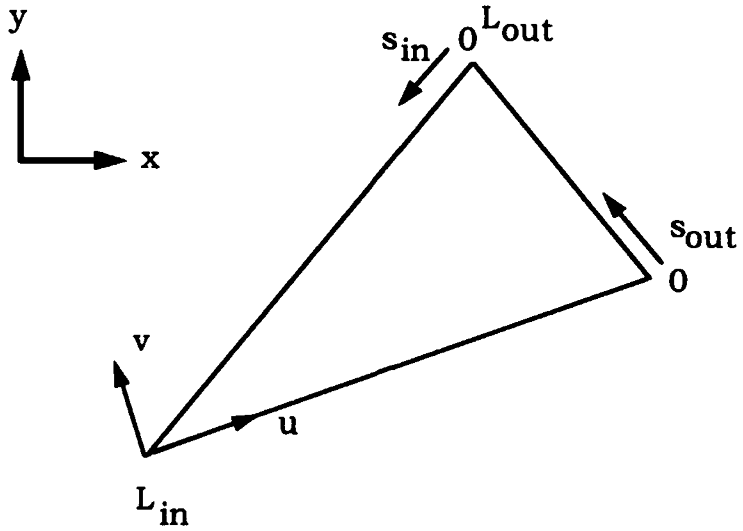


Figure 6. Case 0 Triangle Edge-Moment Orientation

2. Case 0 U-moment Conservation

A u-moment equation for the case 0 triangle can be derived in the same manner as the case 0 cell-balance equation. Equation (84) is multiplied by a linear weight (in this case u) and integrated over the triangle domain as follows:

$$\int_0^h dv \int_{u_L(v)}^{u_R(v)} du\, u \left[\mu' \frac{d\psi(u,v,\mu')}{du} + \sigma_t(u,v)\,\psi(u,v,\mu') = \sigma_s(u,v)\,\phi(u,v) + S_{EXT}(u,v) \right]. \quad (101)$$

Integrating equation (101) produces the following

$$\mu' \left[\left(\frac{u_1}{b} - 1 \right) (\theta_{OUT} + \psi_{OUT}) + \left(\frac{u_1}{b} \right) (\theta_{IN} - \psi_{IN}) + 2\,\psi_{OUT} - \psi_A \right] + \sigma_t\,\psi_U = S_U. \quad (102)$$

Equation (102) is known as the case 0 u-moment conservation equation (with respect to u). The first order u-moment equation differs from its rectangle counterpart in that Legendre polynomial weights were not used for triangle moments. Legendre polynomials lose their orthogonality when integrated on triangular spatial domains and thus lose their utility. The u weight is chosen for simplicity of derivation.

3. Case 0 V-moment Conservation

Similarly, multiplication of equation (84) by the linear weight υ and integration over the triangle produces υ-moment conservation for the case 0 triangle, which, is given by

$$\mu' \frac{\mathrm{h}}{\mathrm{b}} \left[(\theta_{\mathrm{OUT}} + \theta_{\mathrm{IN}}) + (\psi_{OUT} - \psi_{IN})\right] + \sigma_t \, \psi_V = S_V. \tag{103}$$

Equation (103) is referred to as the case 0 υ-moment conservation equation (with respect to a weight of υ).

In these conservation equations, the cell u-moments are defined as

$$\psi_U = \frac{1}{A} \int_{\Delta} dA \; u(\vec{r}) \, \psi(\vec{r}) \tag{104}$$

and

$$S_U = \frac{1}{A} \int_{\Delta} dA \; u(\vec{r}) S(\vec{r}). \tag{105}$$

Similarly, the cell υ-moments are defined by

$$\psi_V = \frac{1}{A} \int_{\Delta} dA \; \upsilon(\vec{r}) \, \psi(\vec{r}) \tag{106}$$

and

$$S_V = \frac{1}{A} \int_{\Delta} dA \; v(\bar{r}) S(\bar{r}) \tag{107}$$

The cell-edge first moments for the input and output edges of the case 0 triangle are given by

$$\theta_{IN} = \frac{1}{L_{IN}} \int_{A} ds_{IN} \; P_1(s_{IN}) \psi(s_{IN}) \tag{108}$$

and

$$\theta_{OUT} = \frac{1}{L_{OUT}} \int_{A} ds_{OUT} \; P_1(s_{OUT}) \psi(s_{OUT}). \tag{109}$$

4. Orientation for First Order Edge-Moments

Orientation is important for higher order edge-moments (zeroth order moments are orientation independent), so a direction convention must be imposed. The triangle edge first moments are defined with a counter-clockwise orientation as shown in figure 7 and are defined as

$$\theta_i = \frac{1}{L_i} \int_{A} ds_i \; P_1(s_i) \psi(s_i), \tag{110}$$

where

$$P_1(s_i) = 2\frac{s_i}{L_i} - 1. \tag{111}$$

C. Rotation/Translation between Coordinate Frames

Since the case 0 coordinate system changes with each angle in the discrete ordinate angle set, a method for moving to and from (u,v) is needed. This is accomplished through rotation and translation of the input and output spatial moments.

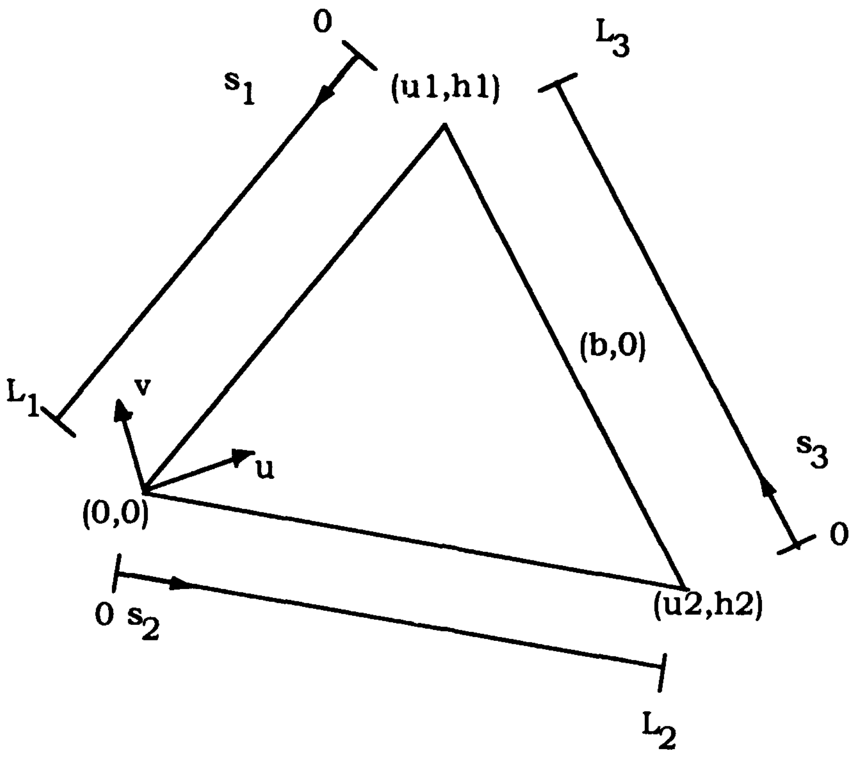


Figure 7. Triangle Edge-moment Direction Convention.

1. Cell Zeroth Order (Average) Moments

A cell zeroth order or average moment for the function $F(x,y)$ is defined by

$$F_A = \frac{1}{A}\int_A dA\, F(\bar{r}) = \frac{1}{A}\int_x dx \int_y dy\, F(x,y). \tag{112}$$

A change of frame by rotation through a counter-clockwise angle Θ produces the following

$$\int_A dA\, F(\bar{r}) = \int_y dy \int_x dx\, F(u(x,y), v(x,y))|J| \tag{113}$$

where J is the determinant of the Jacobian matrix and is given by

$$J = \begin{vmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{vmatrix}. \tag{114}$$

The coordinates u and v are functions of x and y given by

$$u = x\ \cos(\Theta) + y\ \sin(\Theta) \tag{115}$$

$$v = -x\ \sin(\Theta) + y\ \cos(\Theta). \tag{116}$$

Since $|J|$ is unity for all rotations in the plane, the cell-average is unchanged. Similar arguments hold for translations of cell-edge averages. As a result, all cell and cell edge zeroth moments are rotation and translation invariant.

2. First Order Moments

The first order integral moments differ from the zeroth order moments because they introduce a linear weight into the integration of equation (113). This weight causes changes as these moments are rotated and translated and make these moments coordinate frame dependent. In fact, the rotat:on and translation relationships are unique to the weight used.

A cell u-moment for the function $F(x,y)$ is defined by

$$F_U = \frac{1}{A} \int_v dv \int_u du\ u F(u,v). \tag{117}$$

A change of variable to (x,y) by rotating through an angle Θ produces

$$\int_v dv \int_u du\ u\ F(u,v) = \int_y dy \int_x dx\ u(x,y)\ F\big(u(x,y),v(x,y)\big)|J|. \tag{118}$$

Using the definitions of u (equation (115)) and v (equation (116)), and applying the definitions of first moments, the rotation relationship becomes

$$F_U = F_X \cos(\Theta) + F_Y \sin(\Theta). \tag{119}$$

Similarly the v-moment rotation relationship can be obtained by substituting v for u in equation (118) and is given by

$$F_V = F_Y \cos(\Theta) - F_X \sin(\Theta). \tag{120}$$

Translation to another frame can also be achieved by a change of variables. Let x_0 be a distance separating x and x' such that

$$x = x' - x_0. \tag{121}$$

Perform a change of variable on the moment F_X to produce

$$F_X = \frac{1}{A}\int_y dy \int_x dx\; x F(x,y) = \frac{1}{A}\int_y dy \int_{x'} dx'\; x(x') F(x(x'),y). \tag{122}$$

Substituting the definition of x and applying first moments definitions produces

$$F_X = F_{X'} - x_0 F_A. \tag{123}$$

Similarly, a translation of the y' by a distance y_0 produces

$$F_Y = F_{Y'} - y_0 F_A. \tag{124}$$

As a result, counter-clockwise rotations and translations can be combined and summarized into the following set of relationships. Movement from the local (u,v) frame to the global (x,y) is given by

$$F_X = \cos(\Theta) F_U - \sin(\Theta) F_V + x_0 F_A \tag{125}$$

and

$$F_Y = \sin(\Theta)\, F_U + \cos(\Theta)\, F_V + y_0\, F_A \tag{126}$$

where x_0 and y_0 are the x and y coordinates of the (u,v) origin. Inversely, the u and v moments can be written in terms of x and y and are given by

$$F_U = \cos(\Theta)\left(F_X - x_0\, F_A\right) + \sin(\Theta)\left(F_Y - y_0\, F_A\right) \tag{127}$$

and

$$F_V = \cos(\Theta)\left(F_Y - y_0\, F_A\right) - \sin(\Theta)\left(F_X - x_0\, F_A\right). \tag{128}$$

Equations (125) - (128) are relationships specific to the weights u and v (x and y) and are not valid if any other weights are used.

Zeroth and first order edge-moments are not coordinate frame dependent. The integrations which define these moments are constrained to the triangle edges of interest. Rotations and translations to and from local frames have no effect on the orientation of the edge contour variables. As a result, edge-moments are invariant to the rotations and translations discussed above.

3. Splitting of Edge Zeroth and First Moments

The case 1 triangle orientation has only one triangle edge that contains input information. However, the requirement to split into two case 0 triangles to perform transport necessitates a split in the input edge angular flux moments.

Splitting the input edge-moments consists of constructing the angular flux edge distribution for the entire edge and each of the split edges. The input edge distribution can be approximated by

$$\psi(s_{IN}) \approx \psi_{IN} + 3\,P_1(s_{IN})\,\theta_{IN} \tag{129}$$

where $P_1(s_{IN})$ is the Legendre polynomial shifted and scaled along s_{IN} to give

$$P_1(s_{IN}) = 2\frac{s_{IN}}{L_{IN}} - 1.$$

Similarly, distributions can be constructed on each piece of the input edge (after splitting) and are given by

$$\psi_P(s_{IN}) \approx \psi_{IN,P} + 3\,P_1(s_{IN,P})\,\theta_{IN,P} \tag{130}$$

and

$$\psi_N(s_{IN}) \approx \psi_{IN,N} + 3\,P_1(s_{IN,N})\,\theta_{IN,N} \tag{131}$$

where $\psi_{IN,P}$, and $\psi_{IN,N}$, are the edge average moments for the positive $(h > 0)$ and negative $(h < 0)$ oriented case 0 triangles, respectively; $\theta_{IN,P}$, and $\theta_{IN,N}$ are the edge first moments for the positive and negative oriented case 0 triangles; and $P_1(s_{IN,N})$ and $P_1(s_{IN,P})$ are shifted and scaled first order Legendre polynomials on each piece of the input edge. An input edge for a case 1 triangle is shown in figure 8.

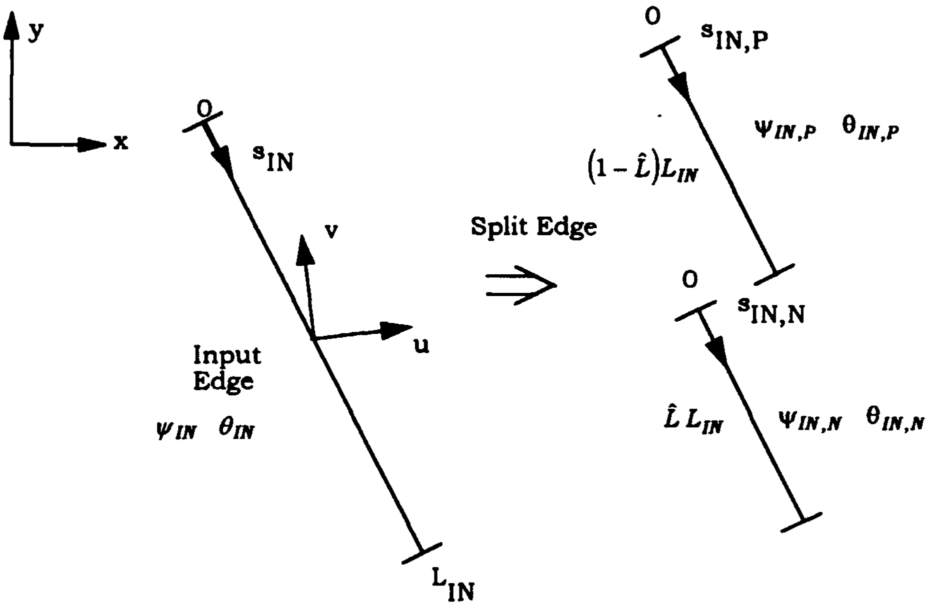


Figure 8. Input Edge-moment Splitting

The goal is to construct two piecewise linear distributions that when integrated on their domains can be summed to retrieve the input edge-moment for the entire input edge. This can be done by applying the input edge-moment definitions to each piece of the split edge and summing them. This produces relationships between the input and split input edge-moments. These relationships are given by

$$\psi_{IN,P} = \psi_{IN} - 3\left(1-\hat{L}\right)\theta_{IN}, \tag{132}$$

$$\psi_{IN,N} = \psi_{IN} + 3\hat{L}\theta_{IN}, \tag{133}$$

$$\theta_{IN,P} = \hat{L}\theta_{IN}, \tag{134}$$

and

$$\theta_{IN,N} = \left(1-\hat{L}\right)\theta_{IN}, \tag{135}$$

where $\hat{L}$ is the fractional length of the negatively oriented triangle input edge over the total input edge length.

4. Assembly of Edge Zeroth and First Moments

The case 2 triangle orientation requires assembly of the output edge angular flux moments. This process is analogous to the moments splitting procedures discussed above. Angular flux distributions are formed for the entire output edge and each edge piece. The split edge pieces are multiplied by the Legendre weights corresponding to the entire edge and integrated piecewise. This produces relationships between the split edge angular flux moments and the assembled moments.

The assembly relationships are given by

$$\Psi_{OUT} = \Psi_{OUT,P}\,\hat{L} + \Psi_{OUT,N}\left(1-\hat{L}\right) \tag{136}$$

and

$$\theta_{OUT} = \hat{L}\left(1-\hat{L}\right)\left(\psi_{OUT,N} - \psi_{OUT,P}\right) + \hat{L}^2\,\theta_{OUT,P} + \left(1-\hat{L}\right)^2\theta_{OUT,N} \tag{137}$$

where $\psi_{OUT,P}$ and $\psi_{OUT,N}$ are the positive and negative oriented triangle zeroth edge-moments along the output edge. Similarly, $\theta_{OUT,P}$ and $\theta_{OUT,N}$ are first moments along the output edge for the positive and negative oriented triangles.

D. Case 0 Spatial Quadratures

Once a coordinate system is chosen, moments definitions are established, and a set of conservation equations are developed, the spatial quadratures can be derived on the case 0 triangle unit cell. For completeness, three triangle spatial quadratures are presented. They are the triangle step, step characteristic, and linear characteristic methods.

1. Case 0 Step Method.

The triangle step method assumes that the average angular flux, ψ_A, within the triangle is equal to the average angular flux along the exiting edge, ψ_{OUT}. This assumption is completely analogous to the rectangle step approximation of last chapter. Using the step assumption and the case 0 cell-balance relationship, ψ_A can be computed directly from equation (93) and becomes

$$\psi_A = \frac{\left(2\,\psi_{IN} + \frac{b\,S_A}{\mu'}\right)}{(2+\varepsilon_U)} \tag{138}$$

and

$$\psi_{OUT} = \psi_A \tag{139}$$

where ε_U is the optical thickness along the base of the case 0 triangle given by

$$\varepsilon_U = \frac{b\sigma_t}{\mu'}. \tag{140}$$

2. Case 0 Step Characteristic.

The triangle step characteristic spatial quadrature assumes that the source distribution S_A is approximately constant throughout the entire triangle. In addition, it also assumes input and output edge angular fluxes can be app^roximated by constants. Using these approximations, the integral Boltzmann neutral particle transport (equation (89)) can be evaluated analytically over the domain of the triangle to obtain the case 0 step characteristic results. These results are given by

$$\psi_A = 2\,\psi_{IN}\,\mathcal{M}_1(\varepsilon_U) + \frac{b\,S_A}{\mu'}\mathcal{M}_2(\varepsilon_U), \tag{141}$$

and

$$\psi_{OUT} = \psi_{IN}\,\mathcal{M}_0(\varepsilon_U) + \frac{b\,S_A}{\mu'}\mathcal{M}_1(\varepsilon_U) \tag{142}$$

where $\mathcal{M}_i(x)$ are the same exponential moment functions defined in the previous chapter.

3. Case 0 Linear Characteristic.

A general triangle linear-nodal hybrid method was developed by Paternoster [14:27]. In that work, a linear characteristic spatial quadrature was performed on case 2 triangles; however, case 1 triangles used the linear nodal method to avoid discontinuities along the output edges. Furthermore, finite difference approximations were used to estimate first order moments for some cases. The finite difference approximations destroy first order conservation. Also, Paternoster tested his method only on equilateral and banded triangle meshes. The method presented here is a pure linear characteristic method that conserves both zeroth and first order moments for all triangle orientations and is tested on arbitrary (even randomly designed) triangle meshes.

The triangle linear characteristic spatial quadrature assumes that the source distribution can be approximated by a linear distribution throughout the entire triangle. Using the three input source moments

S_A, S_X, and S_Y and moment rotation/translation relationships; a linear source distribution can be constructed as follows

$$S(u,v)=a'+b'u+c'v \tag{143}$$

where a', b', and c' are source coefficients dependent on triangle orientation and the input source moments. These coefficients can be obtained directly from the definitions of the source moments (substituting equation (143) into the integral definitions of the moments). For the case 0 orientation the source coefficients are given by

$$a'=\left[\frac{(6(h(3\,b\,S_A-4\,S_U)-4\,S_V\,(b-u_1)))}{A}\right], \tag{144}$$

$$b'=\left[\frac{(24\,(h(2\,S_U-b\,S_A)+S_V\,(b-2\,u_1)))}{(A\,b)}\right], \tag{145}$$

and

$$c'=\left[\frac{\left(24\left(2\,S_V\left(b^2-b\,u_1+u_1{}^2\right)+h\left(S_U\,(b-2\,u_1)+S_A\,b\,(u_1-b)\right)\right)\right)}{A^2}\right], \tag{146}$$

where

$$A=\frac{b\,h}{2}. \tag{147}$$

For a case 1 orientation the source coefficients are given by

$$a'=\left[\begin{array}{c}\dfrac{\left(3\left(3\,A_1{}^2-2\,A_1\,A_2+3\,A_2{}^2\right)S_A\right)}{\left(A^2\right)}-\dfrac{\left(12\,\left(A_1{}^2+A_2{}^2\right)S_U\right)}{\left(A^2\,b\right)}+\\ \dfrac{\left(6\,S_V\left(A_1\,A_2\,b-A_1{}^2\,(b-u_1)+A_2{}^2\,u_1\right)\right)}{\left(A^2\,A_1\right)}\end{array}\right], \tag{148}$$

$$b' = \left[\begin{array}{c} \frac{\left(-12\left(A_1{}^2 + A_2{}^2\right) S_A\right)}{\left(A^2\ b\right)} + \frac{\left(24\left(A^2 - A_1\ A_2\right) S_U\ \right)}{\left(A^2\, b^2\right)} + \\ \frac{\left(6\, S_V\, (A_1 - A_2)\, A_1\, b - 2\left(A^2 - A_1\, A_2\right) u_1\right)}{\left(A^2\, A_1\, b^2\right)} \end{array} \right], \tag{149}$$

and

$$c' = \left[\begin{array}{c} \frac{6}{\left(A^2\, A_1\right)}\left(S_A\left(-A_1\,(A_1 - A_2)b + \left(A_1{}^2 + A_2{}^2\right)u_1\right)\right) + \\ \frac{6}{\left(A^2\, A_1\, b\right)}\left(\left(S_U\left(A_1(A_1 - A_2)b - 2\left(A^2 - A_1A_2\right)u_1\right)\right)\right)\right) + \\ \frac{6}{\left(A^2\, A_1{}^2\right)}\left(S_V\left(A_1{}^2\, b^2 + A_1\,(-A_1 + A_2)\, b\, u_1 + \left(A^2 - A_1\, A_2\right)u_1{}^2\right)\right) \end{array} \right] \tag{150}$$

where A, A_1, and A_2 are triangle areas given by

$$A_1 = \frac{b\, h_1}{2}, \tag{151}$$

$$A_2 = \frac{-b\, h_2}{2}, \tag{152}$$

and

$$A = A_1 + A_2. \tag{153}$$

For a case 2 orientation the source coefficients are given by

$$a' = \left[\frac{\left(3\left(3\, A_1\, b\, S_A - 4\, A_1\, S_U - 2\, b\, S_V\,(b - u_1)\right)\right)}{(A_1\, b\,)} \right], \tag{154}$$

$$b' = \left[\begin{array}{c} \frac{12\left(-A^2\, b\, S_A + 2\left(A^2 - A_1\, A_2\right) S_U\right)}{\left(A^2\, b^2\right)} + \\ \frac{6\, S_V\left(2 u_1\left(A_1\, A_2 - A^2\right) + A\,(2\, A - A_1)\, b\right)}{\left(A^2\, A_1\right)} \end{array} \right], \tag{155}$$

and

$$c' = \left[\begin{array}{c} \dfrac{\left(6\left(A^2\, b\, S_A\,(-b+u_1)+S_U\left(A\,(2\,A-A1)b-2\left(A^2-A_1\,A_2\right)u_1\right)\right)\right)}{\left(A^2\,A_1\,b\right)}+ \\ \dfrac{\left(6\,S_V\left(A^2\,b^2-A(2\,A-A_1)b\,u_1+\left(A^2-A_1\,A_2\right)u_1{}^2\right)\right)}{\left(A^2\,A_1{}^2\right)} \end{array} \right]. \quad (156)$$

Calculating source coefficients is advantageous since they are defined globally over the entire triangle. Even if the general triangle is split into case 0 sub triangles the coefficients for each sub triangle remain unchanged. This property eliminates the need to share cell source moments with sub triangles through splitting.

To derive the zeroth order moments for the case 0 linear characteristic spatial quadrature, equation (89) is integrated across a positively oriented case 0 ($h>0$) triangle. The cell-average flux is given by

$$\psi_A = \frac{1}{A}\int_0^h dv \int_{u_L(v)}^{u_R(v)} du\, \psi(u,v) \quad (157)$$

where

$$u_L(v) = u_1\,\frac{v}{h} \quad (158)$$

and

$$u_R(v) = b-(b-u_1)\frac{v}{h}. \quad (159)$$

Performing the integration produces

$$\psi_A = \left[\begin{array}{c} 2\,\psi_{IN}\,\mathcal{M}_1(\varepsilon_U)+6\,\theta_{IN}\left(\mathcal{M}_1(\varepsilon_U)-\mathcal{M}_2(\varepsilon_U)\right)+ \\ \dfrac{a'\,b\,\mathcal{M}_2(\varepsilon_U)}{\mu'}+\dfrac{b\left(c'\,h+b'(b+u_1)\right)\mathcal{M}_3(\varepsilon_U)}{3\,\mu'} \end{array} \right]. \quad (160)$$

The linear characteristic u-moment is derived in a similar fashion and is given by

$$\psi_U = \frac{1}{A}\int_0^h dv \int_{u_L(v)}^{u_R(v)} du\, u\, \psi(u,v) \tag{161}$$

which produces

$$\psi_U = \psi_U^{\psi_{IN}} + \psi_U^{\theta_{IN}} + \psi_U^{a'} + \psi_U^{b'} + \psi_U^{c'} \tag{162}$$

where

$$\psi_U^{\psi_{IN}} = \psi_{IN}\left[2b\,\mathcal{M}_1^1(\varepsilon_U) + (u_1 - 2b)\mathcal{M}_2^1(\varepsilon_U)\right], \tag{163}$$

$$\psi_U^{\theta_{IN}} = \theta_{IN}\begin{bmatrix} 6b\,\mathcal{M}_1(\varepsilon_U) - 3(4b - u_1)\mathcal{M}_2(\varepsilon_U) + \\ 2(3b - 2u_1)\mathcal{M}_3(\varepsilon_U) \end{bmatrix}, \tag{164}$$

$$\psi_U^{a'} = a'\left[\frac{\left(3b^2\,\mathcal{M}_2(\varepsilon_U)\right) + \left(b(u_1 - 2b)\mathcal{M}_3(\varepsilon_U)\right)}{3\mu'}\right], \tag{165}$$

$$\psi_U^{b'} = b'\left[\frac{\left(2b^2(b + u_1)\mathcal{M}_3(\varepsilon_U)\right) - \left(b\left(b^2 + bu_1 - u_1{}^2\right)\mathcal{M}_4(\varepsilon_U)\right)}{6\mu'}\right], \tag{166}$$

and

$$\psi_U^{c'} = c'\left[\frac{4b^2h\,\mathcal{M}_3(\varepsilon_U) + bh(2u_1 - 3b)\mathcal{M}_4(\varepsilon_U)}{12\mu'}\right]. \tag{167}$$

Derivation of the linear characteristic v-moment differs from the u-moment derivation only by the weight change from u to v. The integration is given by

$$\psi_V = \frac{1}{A}\int_0^h dv\, v \int_{u_L(v)}^{u_R(v)} du\, \psi(u,v) \tag{168}$$

which produces

$$\psi_V = \left[\begin{array}{c} \psi_{IN}\, h\, \mathcal{M}_2(\varepsilon_U) + \theta_{IN} h\, \varepsilon_U\big(\mathcal{M}_4(\varepsilon_U) - \mathcal{M}_3(\varepsilon_U)\big) + \\ \dfrac{\big(4\, a' b\, h\, \mathcal{M}_3(\varepsilon_U) + \big(2\, b\, c' h^2 + b\, b' h(b + 2u_1)\big)\mathcal{M}_4(\varepsilon_U)\big)}{12\,\mu'} \end{array}\right]. \qquad (169)$$

Output edge-moments are derived by parameterizing ψ in terms of the contour variable, s_{OUT}, and integrating as follows:

$$\psi_{OUT} = \frac{1}{L_{OUT}} \int_0^{L_{OUT}} ds\, \psi(s_{OUT}), \qquad (170)$$

where s_{OUT} behaves according to the counter-clockwise convention discussed earlier and L_{OUT} is the length of the output edge of the case 0 triangle. Performing the integration results in the following

$$\psi_{OUT} = \psi_{OUT}^{\psi_{IN}} + \psi_{OUT}^{\theta_{IN}} + \psi_{OUT}^{a'} + \psi_{OUT}^{b'} + \psi_{OUT}^{c'}, \qquad (171)$$

where

$$\psi_{OUT}^{\psi_{IN}} = \psi_{IN}\, \mathcal{M}_0(\varepsilon_U), \qquad (172)$$

$$\psi_{OUT}^{\theta_{IN}} = \theta_{IN}\Big[3\, \varepsilon_U\, \big(\mathcal{M}_2(\varepsilon_U) - \mathcal{M}_1(\varepsilon_U)\big)\Big], \qquad (173)$$

$$\psi_{OUT}^{a'} = a'\left[\frac{\big(b\big(9\,\mathcal{M}_1(\varepsilon_U) - 4\,(3 - \varepsilon_U)\mathcal{M}_2(\varepsilon_U) - 4\, \varepsilon_U\, \mathcal{M}_3(\varepsilon_U)\big)\big)}{\mu'}\right], \qquad (174)$$

$$\psi_{OUT}^{b'} = b'\left[\frac{b(b + u_1)\big(6\,\mathcal{M}_1(\varepsilon_U) - (8 - 3\,\varepsilon_U)\, \mathcal{M}_2(\varepsilon_U) - 3\, \varepsilon_U\, \mathcal{M}_3(\varepsilon_U)\big)}{2\,\mu'}\right], \qquad (175)$$

and

$$\psi_{OUT}^{c'} = c'\left[\frac{b\,h\big(6\,\mathcal{M}_1(\varepsilon_U) - (8 - 3\,\varepsilon_U)\mathcal{M}_2(\varepsilon_U) - 3\,\varepsilon_U\, \mathcal{M}_3(\varepsilon_U)\big)}{2\,\mu'}\right]. \qquad (176)$$

Similarly, the first moment along the output edge is calculated by

$$\theta_{OUT} = \frac{1}{L_{OUT}} \int_0^{L_{OUT}} ds\, P_1(s_{OUT})\psi(s_{OUT}) \tag{177}$$

which gives

$$\theta_{OUT} = \theta_{OUT}^{\psi_{IN}} + \theta_{OUT}^{\theta_{IN}} + \theta_{OUT}^{a'} + \theta_{OUT}^{b'} + \theta_{OUT}^{c'} \tag{178}$$

where

$$\theta_{OUT}^{\psi_{IN}} = \psi_{IN}\, \varepsilon_U \left(\mathcal{M}_1(\varepsilon_U) - \mathcal{M}_2(\varepsilon_U)\right), \tag{179}$$

$$\theta_{OUT}^{\theta_{IN}} = 3\,\theta_{IN} \left(4\left(\mathcal{M}_1(\varepsilon_U) - \mathcal{M}_2(\varepsilon_U)\right) - \mathcal{M}_0(\varepsilon_U)\right), \tag{180}$$

$$\theta_{OUT}^{a'} = a' \left[\begin{array}{c} \dfrac{(-5b - 4u_1)\mathcal{M}_1(\varepsilon_U) + (b(11 - 2\varepsilon_U) + 2(5 - \varepsilon_U)u_1)\mathcal{M}_2(\varepsilon_U)}{\mu'} + \\ \dfrac{-2(8 - 5\varepsilon_U)(b + u_1)\mathcal{M}_3(\varepsilon_U) - 4\varepsilon_U(b + u_1)\mathcal{M}_4(\varepsilon_U)}{3\mu'} \end{array} \right], \tag{181}$$

$$\theta_{OUT}^{b'} = b' \left[\begin{array}{c} \dfrac{1}{\mu'}\left(-2\left(b^2 + u_1 + u_1^2\right)\mathcal{M}_1(\varepsilon_U)\right) + \dfrac{1}{2\mu'}\left(2(5 - \varepsilon_U)u_1^2 + \right. \\ \left. b(9 - 2\varepsilon_U)(b + u_1)\mathcal{M}_2(\varepsilon_U)\right) + \\ \dfrac{1}{3\mu'}\left(b^2(7 - 5\varepsilon_U) + b(6 - 5\varepsilon_U)u_1 + \right. \\ \left. (8 - 5\varepsilon_U)u_1^2)\mathcal{M}_3(\varepsilon_U)\right) - \dfrac{1}{3\mu'}\left(2\varepsilon_U\left(b^2 + bu_1 + u_1^2\right)\mathcal{M}_4(\varepsilon_U)\right) \end{array} \right], \tag{182}$$

$$\theta_{OUT}^{c'} = c' \left[\begin{array}{c} \dfrac{\left(-h(b + 2u_1)\mathcal{M}_1(\varepsilon_U)\right)}{\mu'} + \dfrac{(b(4 - \varepsilon_U)h + 2(5 - \varepsilon_U)h\, u_1)\mathcal{M}_2(\varepsilon_U)}{2\mu'} + \\ \dfrac{(-b(4 - 5\varepsilon_U)h - 2(8 - 5\varepsilon_U)h\, u_1)\mathcal{M}_3(\varepsilon_U)}{6\mu'} + \\ \dfrac{\left(\varepsilon_U h(b + 2u_1)\mathcal{M}_4(\varepsilon_U)\right)}{3\mu'} \end{array} \right] \tag{183}$$

E. Triangle Mesh Development and Refinement

The goal of mesh development was to create triangles that are as near as possible to equilateral triangles. As a result, all initial triangle meshes were generated using Mathematica. This product contains a Delaunay triangulation algorithm. The Delaunay triangulation algorithm is discussed briefly below.

Let $N_{i,j}$ be the set of all points in the plane such that

$$N_{i,j} = \{\text{points closer to node } i \text{ than to node } j\}. \tag{184}$$

Now define P_i as the intersection of all $N_{i,j}$ for $j \neq i$ given by

$$P_i = \bigcap_{i \neq j} N_{i,j}. \tag{185}$$

The set P_i is a polygon surrounding node i , containing the points of the plane for which node i is the closest node. Each edge of this polygon is a segment of the perpendicular bisector of the line connecting node i and one of its "nearest neighbor nodes," say node k. The nearest neighbor nodes are listed in counter-clockwise order around the polygon. The Mathematica routine generates such lists. For example, the list of neighbors of node 3 might be {{3},5,9,6,2,4}. This is used to produce a list of triangles, identified by triplets of node indices; for the above example, the triangle list would be {{3,5,9},{3,9,6},{3,6,2},{3,2,4},{3,4,5}}. The meshes produced in this way avoid producing long, narrow triangles.

Delaunay triangulation is a workable means of generating course meshes. Finer meshes were produced by refining the Delaunay meshes by bisecting each triangle edge and connecting the three new indices. This technique gives an effective refinement factor of four and produces triangles that are similar to the parent triangle. Figure 9 shows an arbitrary triangle with the dashed line denoting the refinement.

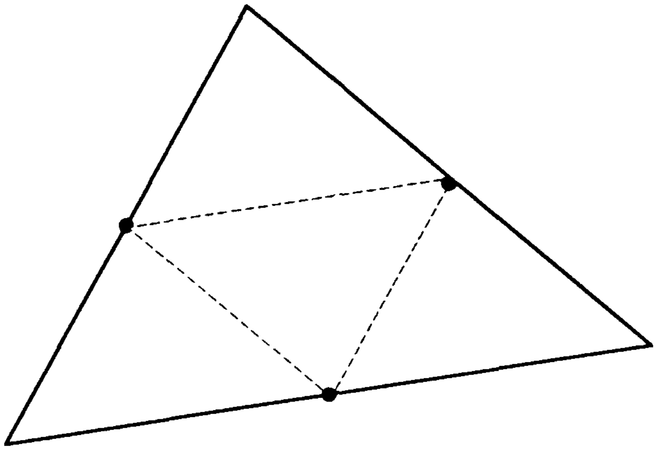


Figure 9. General Triangle Mesh Refinement

IV. General Triangle Algorithm (TRISN)

To test the general triangle linear characteristic spatial quadrature, a general triangle algorithm was developed. I hereafter refer to this algorithm as TRISN. TRISN was implemented in FORTRAN 77 and is discussed in detail below. TRISN is compared to the rectangle discrete ordinates algorithm SNXY. SNXY is also discussed below. The TRISN source code is not provided in this document but a copy is archived at AFIT.

A. The SNXY Rectangle Discrete Ordinates Algorithm

SNXY is a rectangle spatial mesh discrete ordinates algorithm. It contains the same angular quadratures as TRISN and has the rectangle linear characteristic spatial quadrature implemented and benchmarked. SNXY is used as a comparative tool to demonstrate the similarities and differences between TRISN and rectangle spatial mesh discrete ordinate algorithms. The following is a pseudocode outline of the major components of the SNXY algorithm (quadrant one angle set only).

```
Read Problem Definition From File

Initialize Problem

Input Angle and Spatial Quadrature Parameters

Initialize Problem Boundaries

Set previous iteration's scalar flux/moments to this
iteration

DO
```

```
    FOR each angle in the angular quadrature set
        Initialize angle angular flux/ moments to this iteration
        FOR each column of spatial mesh (in order)
            FOR each row of spatial mesh (in order)
                Handle input boundary (beginning of row or column only)
                Assign inputs for cell(column, row)
                Construct input source moments from previous iteration's flux
                Perform transport across cell
                Accumulate angular flux/ moments and particle current/ moments in weighted sums
                Handle output boundary (end of row or column only)
                Assign inputs for neighbor cell(s)
            NEXT row
        NEXT column
    Next angle
LOOP Until converged
Output results
END
```

B. The TRISN Triangle Discrete Ordinates Algorithm

To clearly understand the operation of TRISN, an explanation of the different coordinate frames is needed. First, the global (x,y) frame is the frame to which all mesh nodes are referenced. This frame is established by input and is invariant. The second frame discussed is local (x,y) frame. The local (x,y) frame has axes oriented in the same direction as the global (x,y) frame with origin placement at the centroid of the triangle of interest. All moments are kept at the local (x,y) origin to preclude large translations. The last frame is the local (u,v) frame. The local (u,v) frame is oriented according to the rules established in the last chapter. This frame changes with triangle shape, size, orientation, and angle of transport. This frame is where all transport takes place. All input moments must be rotated and translated to this frame from the local (x,y) frame and all output moments must be rotated and translated out of this frame back to the local (x,y) frame.

The TR!SN triangle discrete ordinates algorithm has many similarities to standard rectangle mesh algorithms. Implementing features such as additional angle quadratures, multi-group energy structure, anisotropic scattering, time dependence, etc., are nearly the same. The significant difference between the triangle approach and rectangle mesh approaches is in handling the inner structure (cell-to-cell transport along a given direction). Rectangle algorithms take advantage of problem symmetries to use looping structures that march across rows and columns to transport particles. However, the data structures required to construct spatial meshes with arbitrary triangle orientations do not support a loop structure. Only banded triangle meshes have features that can take advantage of looping structure. TRISN uses a push-down stack structure which will be discussed later.

The TRISN algorithm starts by reading the spatial grid information. The following is a list of required information for TRISN:

1) an indexed list (location in the list determines index of coordinates pair) list of node coordinate locations in global (x,y) coordinates,

2) an indexed list (location in the list determines triangle index number) of triangle index triplets (e.g. {1,2,3} denotes that coordinate pairs 1, 2, and 3 define the triangle of interest),

3) an indexed triangle to region list (position in list denotes triangle number, value at that location denotes region for triangle),

4) an indexed list of region to material mappings (position denotes region number, value at that location denotes material number for the region of interest), and

5) a list of material properties (scatter cross-section, total cross-section, and flux independent sources).

Once the data is read from the input file, the angular quadrature is chosen. Using the choice of angular quadrature, direction cosines and angle weights are constructed. Next, the spatial quadrature is chosen. Currently implemented choices are the step, step characteristic, and linear characteristic methods. Also, an iteration-to-iteration convergence criterion is set to stop the iterative process.

Iteration zero initialization of scalar flux, scalar flux moments, current, and current moments is accomplished.

Arrays that contain the previous iteration's scalar flux values are set to the current iteration's values. If this is the first iteration, the values are set either to zero or to an initial guess. The current iteration scalar flux, scalar flux moments, particle current, and particle current moment arrays are then initialized for the next angle sweep through the problem. The angular flux and angular flux moment arrays are reinitialized. An angle is chosen from the angular quadrature set. Arrays containing the vector dot product between $\hat{\Omega}$ and triangle edge outward unit normal and input edge flags (one indicating input edge, zero indicating non-input edge) are calculated for the present angle and data available flags arrays are initialized for each edge.

Exterior edge data ready flags for the edges with input boundary data are set to zero (zero indicating data, one indicating no data). Each triangle along the boundary is queried to check if it has the necessary input information to perform a transport calculation (all input edges contain data). This is accomplished by taking the vector dot product of the input edge array and the data ready array. A value of zero indicates the triangle has the necessary inputs; anything else indicates it does not. The index of each triangle that is ready to compute is placed on a stack and the stack pointer is incremented. Once each triangle containing a boundary is queried the first transport calculation is performed.

A triangle index number is pulled from the stack of ready triangles and the stack pointer is decremented. Source moments for the current triangle are constructed by summing external (flux independent) source components (all external source moments except zeroth moment are assumed equal to zero in this implementation) with the scattering components. In general, the source components are given by

$$S_A(itri) = \sigma_s(itri)\phi_A(itri, iter-1) + S_{A_{EXT}}(itri), \tag{186}$$

$$S_X(itri) = \sigma_s(itri)\phi_X(itri, iter-1), \tag{187}$$

$$S_Y(itri) = \sigma_s(itri)\phi_Y(itri, iter-1), \tag{188}$$

where *itri* is the current triangle; *iter* is the current iteration; and σ_s is the scattering cross-section for triangle *itri*; S_A, S_X, and S_Y are the source average, x, and y-moments for triangle *itri*; and $S_{A_{EXT}}$ is the flux independent average source for triangle *itri* (dependent on material). $\boldsymbol{\phi}_A$, $\boldsymbol{\phi}_X$, and $\boldsymbol{\phi}_Y$ are the scalar flux average, x, and y-moments for triangle *itri* which are calculated from integrated previous iteration angular flux moments for triangle *itri*.

Once the source moments are known, the orientation of *itri* and the input and output edges are determined. The results are sorted and stored in the permutation variables IA, IB, and IC by using the following rules:

1) If the orientation of triangle *itri* is case 0, then the variable ICASE is set to zero. Next, IA is set to the input side index number, IB is set to the output side index number, and IC remains unchanged (contains no information).

2) If the orientation of triangle *itri* is case 1 then the variable ICASE is set to one, IA is set to the input side index number, and IB and IC are each set to distinct output side index numbers.

3) If the orientation of triangle *itri* is case 2 then the variable ICASE is set to two, IA and IB are each set to distinct input side index numbers, and IC is set to the output side index number. Next, source moments are calculated and transport on the triangle is performed. The following discussion will center on the linear characteristic spatial quadrature.

The linear characteristic algorithm follows three distinct paths dependent on the triangle orientation. If the triangle is a case 0 triangle, the local axis origin location, (x_0, y_0) is found (rules for origin placement are discussed in the last chapter). Additionally, the height (h), base (b), and u-location of the off axis apex (u_1) are found (see figure 3). Then the source moments S_X and S_Y are rotated and translated from the local (x, y) frame to the local (u,v) frame to form new source moments S_U and S_V. The rotation and translation relationships were derived in the last chapter. The rotated and translated integral source moments S_U and S_V are given by

$$S_U = (S_X - S_A\, x_0)\cos(\Theta) + (S_Y - S_A\, y_0)\sin(\Theta) \tag{189}$$

and

$$S_V = (S_Y - S_A\, y_0)\cos(\Theta) - (S_X - S_A\, x_0)\sin(\Theta) \tag{190}$$

where Θ is the counter-clockwise rotation angle in the plane.

Once the source moments are rotated to (u,v), the source coefficients a', b', and $c'$are calculated dependent on the triangle type and relationships of last chapter.

If the triangle height is negative, then a transformation is accomplished to flip the triangle across the u-axis by negating the v-

moment, changing the sign of c', and changing the sign of the edge input first moment (θ_{IN}). This has the net effect of transforming from v to a new variable v' that has the positive (left to right transport and $h>0$) case 0 orientation desired. Transport is then performed on the triangle using the linear characteristic relationships. Next, the transformation is made back to the negative orientation by negating the v-moment and reversing the sign of the output edge first moment, θ_{OUT}. If the triangle is already positively oriented, then the moments are calculated directly from the linear characteristic relationships.

Once the output moments are calculated in (u,v) they are rotated back into the local (x,y) frame by using the following transformations

$$\psi_X = \psi_U \,\mathrm{Cos}(\Theta) - \psi_V \,\mathrm{Sin}(\Theta) + x_0\, \psi_A \tag{191}$$

and

$$\psi_Y = \psi_U \,\mathrm{Sin}(\Theta) + \psi_V \,\mathrm{Cos}(\Theta) + y_0\, \psi_A . \tag{192}$$

The case 1 orientation differs from the case 0 orientation. The case 1 algorithm starts by splitting the case 1 triangle into one positively oriented case 0 triangle and one negatively oriented case 0 triangle. Then sub-triangle parameters for each triangle (i.e. heights, bases, side lengths, etc.) are calculated. The source coefficients are calculated using the case definitions of last chapter. The input edge zeroth and first moments are split.

Transport is performed on the positive-oriented case 0 triangle. Transformations are performed on the negative-oriented case 0 triangle to transform it into a positive case 0 triangle. Transport is performed and the triangle is transformed back into a negative oriented triangle.

The angular flux cell u, v and zeroth order moments are then assembled to produce moments defined on the entire case 1 triangle. The case 1 angular flux u and v-moments are then rotated back into the local (x,y) frame. The edge-moments are stored in the appropriate array locations.

Case 2 oriented triangles are handled by constructing the local (u,v) frame and calculating the location of the origin. Cell parameters such as sub-triangle heights, bases, edge lengths, and areas are then found. The source moments are rotated to (u,v) and source coefficients are calculated using case 2 relationships of last chapter. Transport is performed on both sub triangles utilizing the necessary transformations to accommodate the negative orientation of one of the sub triangles. Sub triangle angular flux cell-average, u, v, and all output edge-moments are assembled and scaled back to the entire case 2 triangle. After the edge-moments are assembled, the angular flux, u, and v-moments are rotated back to the local (x,y) frame.

Once transport across the whole triangle is performed, the output moments are stored in the appropriate arrays and the exiting edge data is passed to the triangle that shares the edge of interest with this triangle (neighbor triangle) via a cross-reference table (constructed when each new angle is chosen). Each output edge first moment sign is reversed to preserve the counter-clockwise direction convention as it becomes an input edge-moment of the adjacent spatial cell. Next, the input data flags for the neighbor triangle(s) is (are) set to indicate that data is present. The neighbor triangle(s) is (are) then queried to check readiness for transport and the stack and stack pointer are adjusted as necessary.

The angular flux components for the current triangle are weighted and accumulated as the scalar flux moments, per equation (9).

The triangle transport process is repeated until the pending triangle stack is empty (the stack pointer points to zero). At this point, all the angular flux/moments and edge angular flux/moments are present for the current angle.

Once all triangles have been used for the present angle, the next angle in the angle quadrature set is used. The process is repeated, accumulating angular flux/moments and current/moments, until all the angles have been done. At that point, the scalar flux zeroth moments are compared to the previous iteration results. If convergence has not yet been achieved, the outer loop repeats until convergence is achieved.

The following is pseudocode for the TRISN algorithm:

```
READ problem information from file
Initialize angular and spatial quadrature
Set convergence criteria
DO
    Set previous iteration scalar flux arrays to this iteration's values
    Reinitialize scalar flux/moments/current
    FOR each angle in the quadrature set
        Reinitialize angular flux/moments
        Determine which sides are input and output edges
        Update input edge angular flux from boundary data
```

```
Find triangles with boundary data that are ready to
compute and place on the triangle pending stack

IF pointer > 0

    Choose triangle, decrement pointer

    Construct source moments

    Determine the type triangle (which case)

    IF case 0

        Construct local $(u,v)$ frame

        Rotate source moments to $(u,v)$

        Calculate source coefficients $a'$, $b'$, and $c'$

        IF negative case 0 triangle

            Flip triangle

            Transport across triangle

            Flip triangle back

        ELSE

            Transport across triangle

        ENDIF

        Rotate output moments back to local $(x,y)$

    ELSEIF case 1

        Construct local $(u,v)$ frame

        Rotate source moments to $(u,v)$

        Calculate source coefficients $a'$, $b'$, and $c'$
    Split input edge-moments
    Perform transport across positive case 0 triangle
    Invert negative case 0 triangle
    Perform transport on case 0 triangle
    Invert triangle
    Assemble the cell-average and first moments from the case 0 triangles
    Rotate the output moments back to local $(x,y)$
ELSEIF case 2
    Construct local $(u,v)$ frame
    Rotate source moments to $(u,v)$
    Calculate source coefficients $a'$, $b'$, and $c'$
    Perform transport across positive case 0 triangle
    Flip negative case 0 triangle
    Perform transport on case 0 triangle
    Flip triangle back
    Assemble the cell-average and first moments from the case 0 triangles
    Assemble output edge zeroth and first moments
    Rotate moments back to local $(x,y)$
```

```
                ENDIF

                Update neighbor triangles (adjacent to output
                edges)

                Accumulate scalar flux and current

            ENDIF

        NEXT angle

        Check convergence

    LOOP Until converged

    Output results
```

V. Testing

The triangle mesh transport methods developed here were tested on a variety of problems. Although the step and step characteristic methods were successful, the linear characteristic method was most effective. This section focuses on the testing of the linear characteristic method for triangle meshes, as implemented in my TRISN code.

TRISN was benchmarked against SNXY on a variation of Lathrop's square-in-a-square problem [1:312]. Comparisons are made between the rectangle mesh discrete ordinates transport code, SNXY, and the triangle code, TRISN. Rate of convergence calculations are performed to compare the rectangle linear characteristic spatial quadrature to the general triangle linear characteristic quadrature for Lathrop's problem. Mesh sensitivity calculations are performed to quantify the sensitivity to random variations in the triangle spatial mesh. These tests validate the method and the code. Finally, a series of test cases is analyzed to demonstrate the flexibility of general triangle spatial meshing on problems that are not readily represented on rectangular meshes. These tests show the power of the new method.

A. Benchmarking TRISN with Lathrop's Problem

The benchmark process to validate the operation of the TRISN algorithm consisted of comparison of average scalar flux results from SNXY and TRISN on a variation of Lathrop's square-in-a-square problem. In addition, average current and current distributions are compared at the top edge of the problem. The particle current comparisons are made to ensure that the proper amount of particle flux is being transmitted out of the problem.

The benchmark problem consisted of a 2 x 2 square source region centered in a 4 x 4 square. Both regions used a total cross-section of 0.75 and a scattering-to-total cross-section ratio of 2/3. The center source region contained a source of one. All exterior problem boundaries were vacuum with no incident currents. The benchmark problem is shown in figure 10.

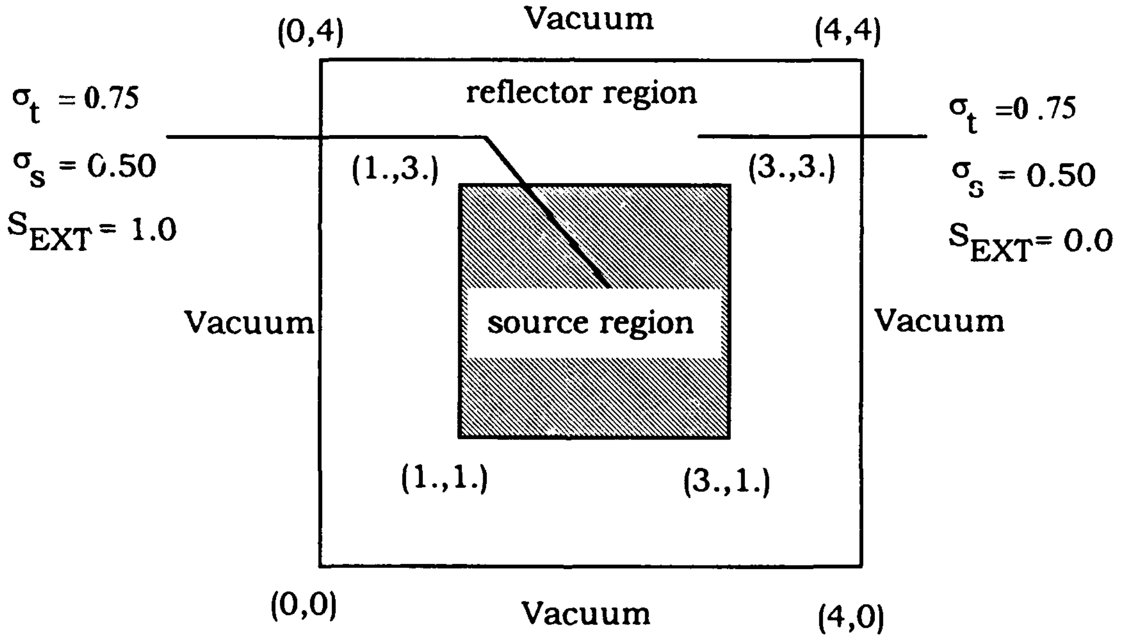


Figure 10. Lathrop's Benchmark Problem for TRISN.

A reference solution was calculated using SNXY on a mesh of 16384-square cells with diamond difference spatial quadrature and S8 level symmetric angular quadrature.

The initial triangle mesh was generated with Delaunay triangulation and is shown in figure 11. Triangle mesh refinements were accomplished using the edge bisection technique discussed earlier. Average scalar flux for each mesh was compared using various spatial mesh refinements. If TRISN is operating properly, the average flux must asymptotically converge to the same result as SNXY as the spatial meshes are refined. The results of these comparisons are shown in figure 12. The dashed line in figure 12 denotes the value of the 16384-cell diamond difference calculation. Figure 12 also shows that as the square and triangle meshes are refined, the average scalar flux results of TRISN and SNXY can be driven as close to the same result as desired. The difference between the flux results of TRISN and SNXY for the most highly resolved meshes on each is less than 0.003 percent.

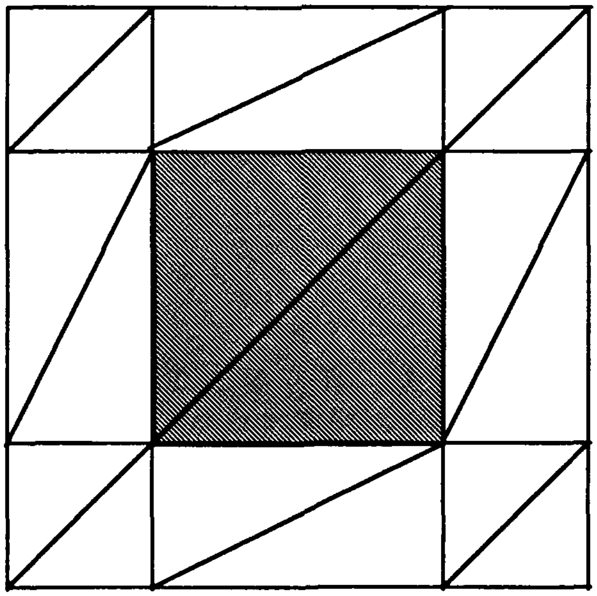


Figure 11. Initial Triangle Mesh for Lathrop's Problem

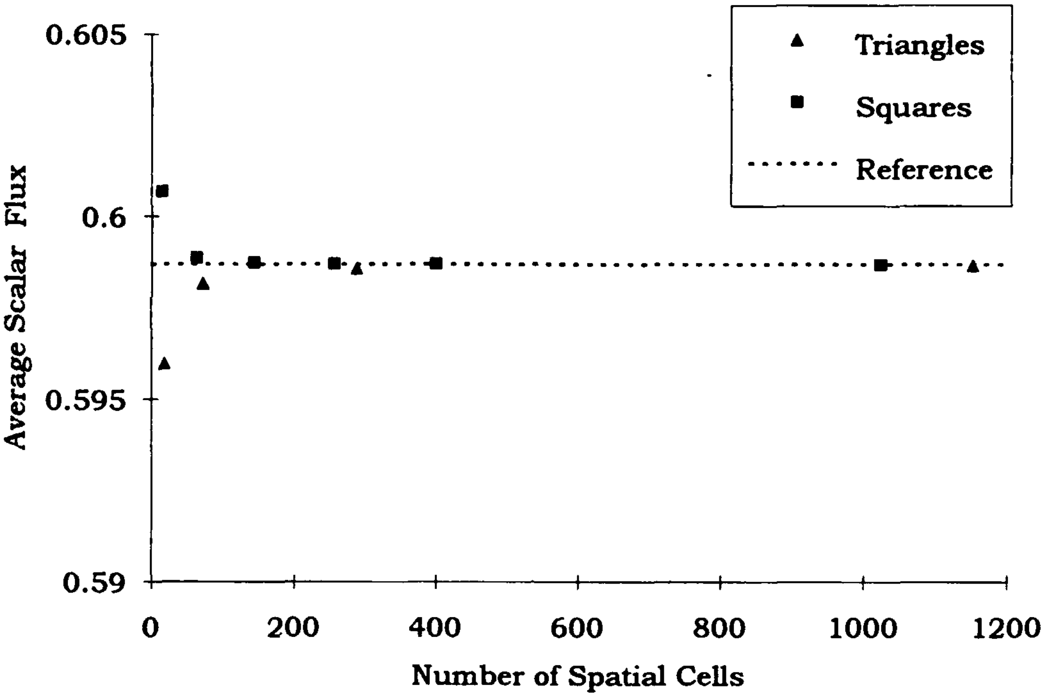


Figure 12. Lathrop's Problem Benchmark Results for TRISN vs. SNXY

The top edge currents for SNXY and TRISN are compared using a 16 x 16 cell square mesh. Each square spatial cell was divided on its diagonal to produce a 512-cell triangle mesh. Figure 13 shows the TRISN and SNXY calculated average outward partial current for each spatial cell along the top edge of the benchmark problem. Figure 13 indicates that SNXY and TRISN implementations of the linear characteristic spatial quadrature are in agreement for the top edge current distribution. (The triangles are present on the graph, but are hidden by the superimposed squares).

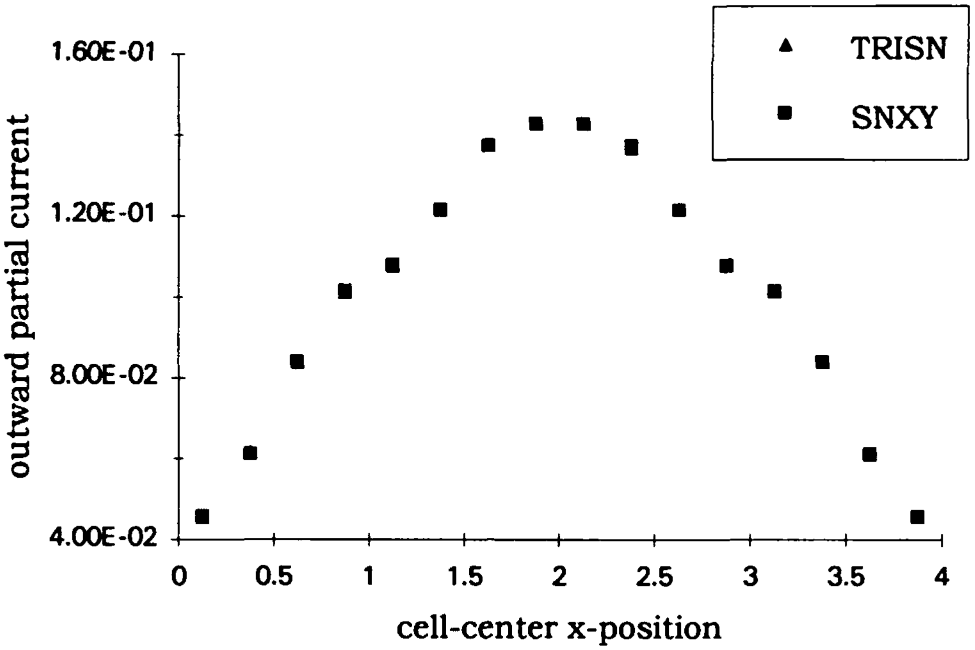


Figure 13. Top Edge Partial Current for Benchmark Problem

The relative difference between the partial currents can be seen by using the following difference ratio

$$E_r = \frac{2\left|j_{SNXY} - j_{TRISN}\right|}{\left|j_{SNXY} + j_{TRISN}\right|}. \tag{193}$$

Figure 14 shows the relative difference of outward partial currents calculations between square and triangle linear characteristic for each spatial cell. The maximum difference in outward partial current is less than 0.2 percent for cells on the order of 1/3 of a mean free path in thickness.

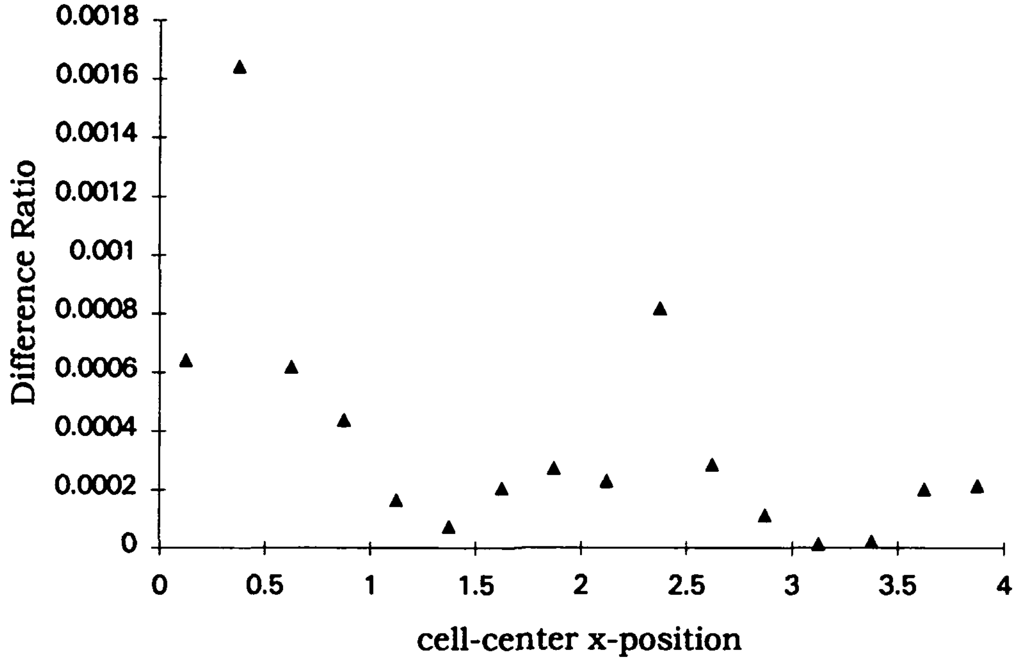


Figure 14. Top Edge Current Relative Difference of Squares vs. Triangles for Benchmark Problem

B. Convergence Rate of Triangle Linear Characteristic

Larsen showed analytically that the asymptotic rate of convergence for rectangle linear characteristic is at least $O(\Delta x^3)$; his numerical experiments gave convergence rates as high as $O(\Delta x^4)$ [6:80]. The triangle linear characteristic convergence rate was measured numencally and compared to the rectangle linear characteristic rate using Lathrop's problem shown in figure 10. Problem scalar flux was calculated for successive refinements of the rectangle and triangle meshes. The relative errors are plotted versus total number of cells on a log/log scale in figure 15. Plotting the information in this way and calculating the slope of the line provides the approximate order of convergence. Figure 15 shows that rectangle linear characteristic and triangle linear characteristic both converge at about $O(\Delta x^3)$ with squares having a slope of -3.2 and triangles a slope of about -2.8. (Note that the linear measure of triangle cell size varies inversely with the square root of the number of cells, which is used in figure 15 and subsequent similar figures.) Convergence rate for thin cells is valuable but a more important issue is method performance in the one mean free path cell regime. Cell thicknesses much less than one mean free path can cause prohibitively large memory constraints on large problems and do not exploit the accuracy of the spatial quadrature method. The test cases that follow will demonstrate performance in the one mean free path regime.

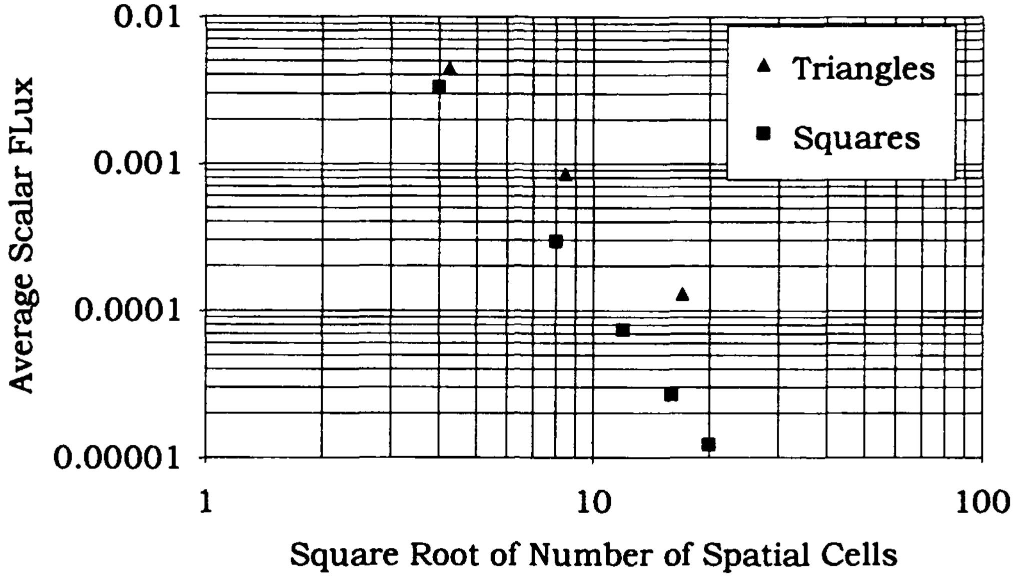


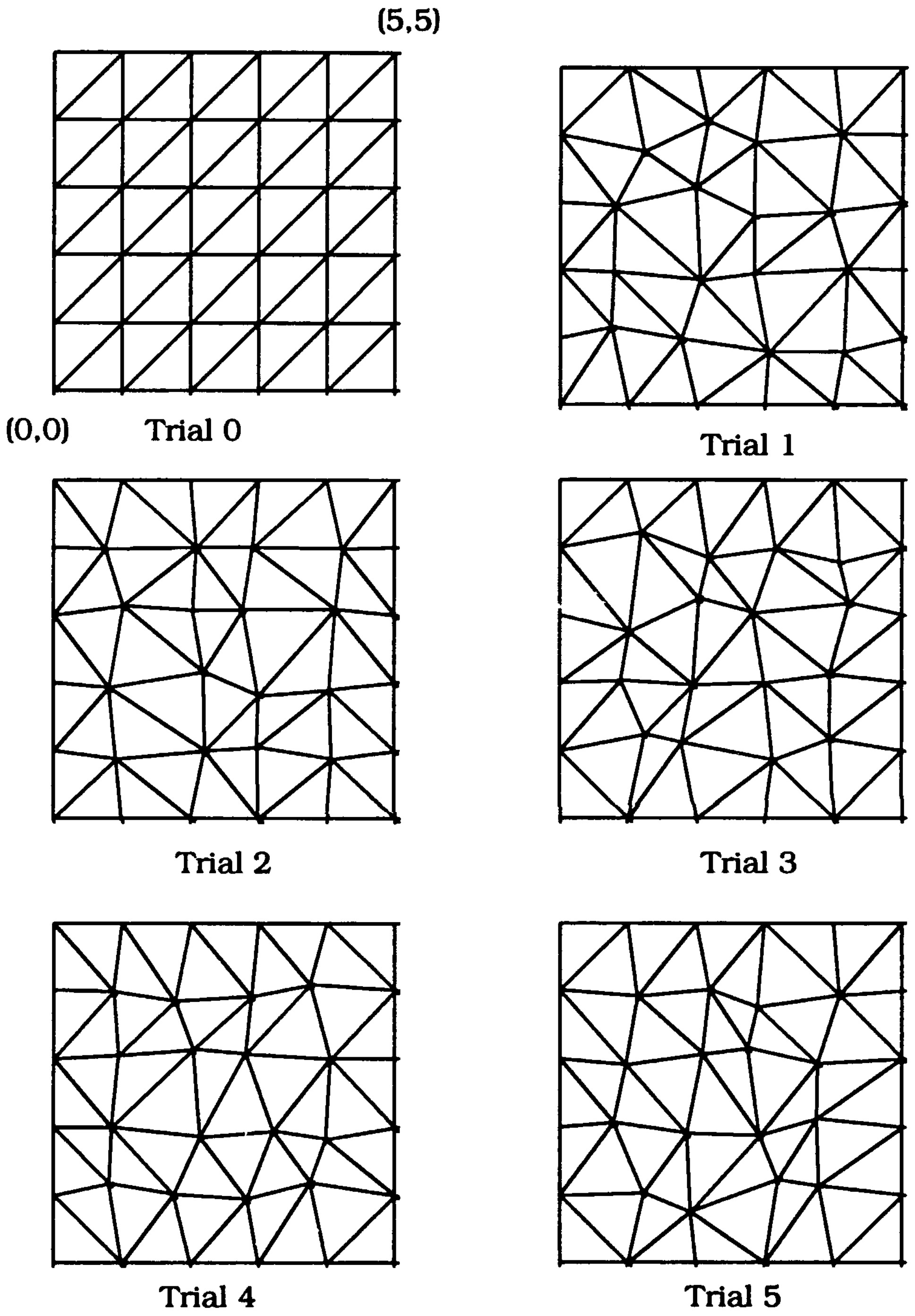
Figure 15. Average Scalar Flux Convergence Rate for Rectangle/Triangle Linear Characteristic.

C. Sensitivity of Linear Characteristic to Triangle Mesh Variation

If the flux and current calculations are sensitive to the spatial mesh node placements for arbitrary triangle meshes, then the utility of general triangle meshing is diminished. Modest variations in node placement should translate to only slight variations in results.

A five mean free path square test problem was used to test the sensitivity of triangle linear characteristic results to random variations in the triangle mesh. This square was a pure absorber with total cross-section 0.75 and a source of 1.0. Triangle meshes were randomly generated (with fixed exterior nodes) and average scalar fluxes were compared to rectangle results on a 5 x 5 square mesh.

Average scalar flux within the square was calculated using an S8 level symmetric angular quadrature. Seven cases were studied. These cases consisted of the square spatial grid (5 x 5 mesh), a triangle grid derived by splitting the square grid along each cell's diagonal (from lower left to upper right in each square), and five cases where the grid interior nodes were varied randomly by 0.25 units in both x and y (from the patterned triangle mesh). For the triangle meshes of figure 16, the node connections were broken by reapplying the Delaunay algorithm on the nodes. In all cases the same number of spatial cells was used and the boundary spatial cell nodes remained fixed. Figure 16 shows the mesh test set. Figure 17 shows the relative difference between each triangle mesh trial and the square mesh computation. Figure 17 shows that trial 0 gives the largest difference from rectangle computations but remains less than .04 percent relative difference from the square mesh computation.



Trial 0

Trial 1

Trial 2

Trial 3

Trial 4

Trial 5

Figure 16. Mesh Sensitivity Test Set for the Sensitivity Test Problem for 0.25 Unit Random Variation

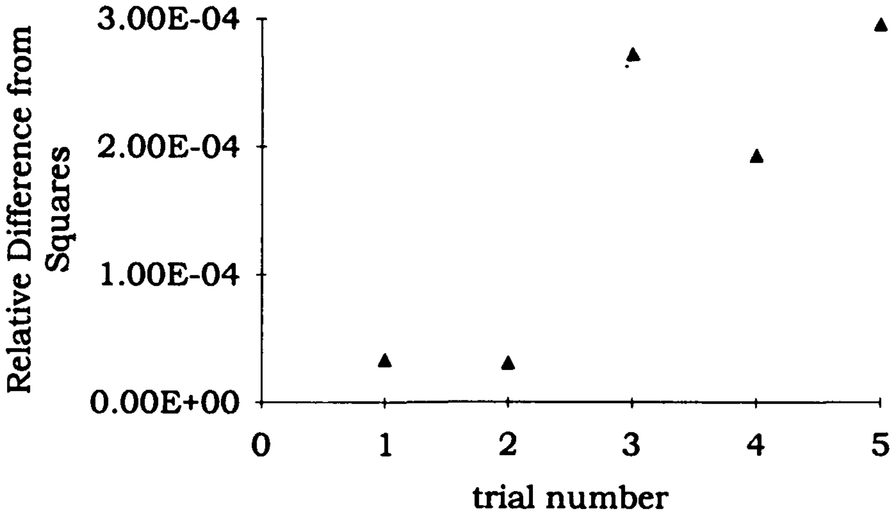


Figure 17. Relative Difference in Scalar Flux for Triangle Linear Characteristic

Top edge particle current comparisons by cell were made as another measure of the sensitivity of the triangle linear characteristic method. The results are shown in Table 1. The relative difference between rectangle calculations and each triangle trial is shown for each spatial cell in figure 18. Figure 18 shows that the maximum relative difference between the triangle linear characteristic and square linear characteristic methods is about 0.03 percent.

Table 1
Top Edge Current for 0.25 Unit Random Variation in Node Placement for the Sensitivity Test Case

x	Squares	Trial #0	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5
0.5	0.20780	0.20785	0.20853	0.20870	0.20798	0.20798	0.20848
1.5	0.24622	0.24632	0.24635	0.24599	0.24619	0.24633	0.24638
2.5	0.25018	0.25021	0.25027	0.25028	0.25032	0.25016	0.25019
3.5	0.24622	0.24639	0.24634	0.24622	0.24625	0.24611	0.24634
4.5	0.20780	0.20899	0.20790	0.20791	0.20799	0.20825	0.20804

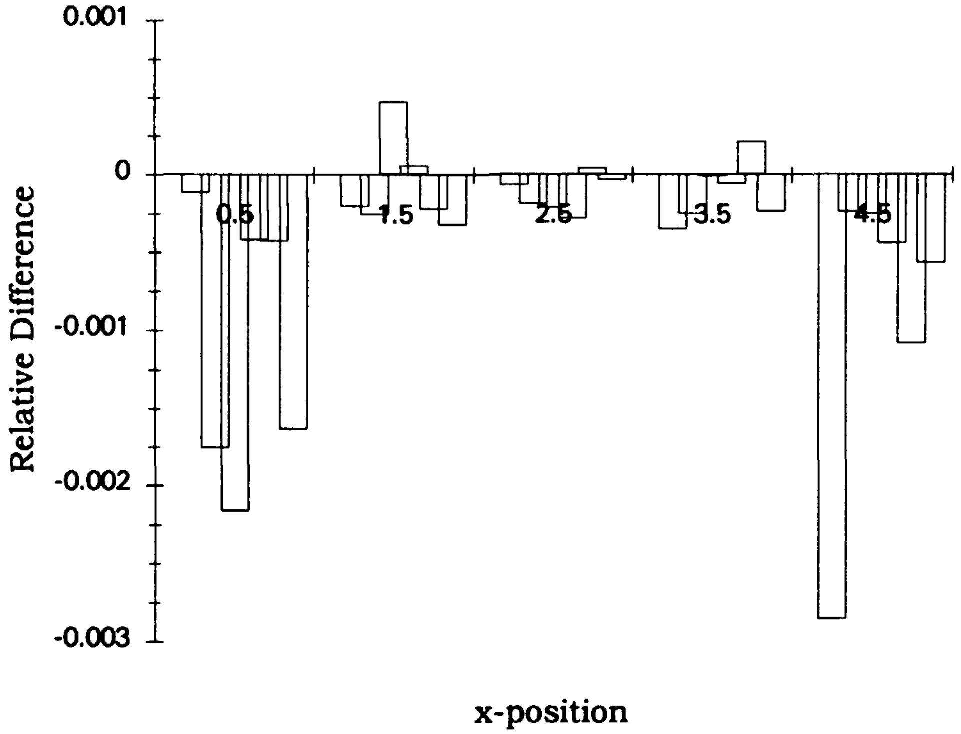
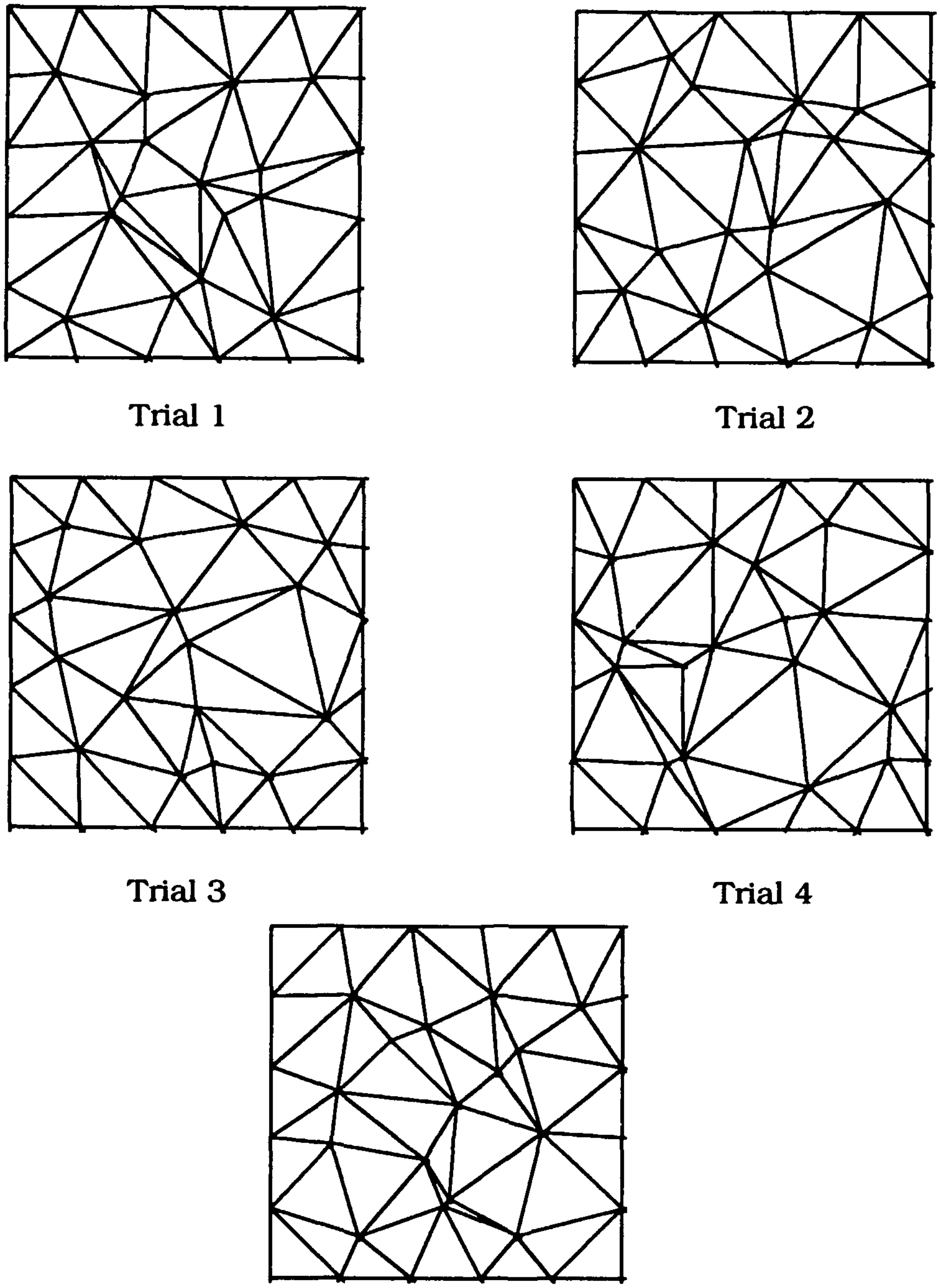


Figure 18. Top Edge Current Relative Difference From Square Calculations for Sensitivity Problem for 0.25 Unit Random Variation in Node Placement

The sensitivity analysis was repeated using a 0.50 unit random variation in internal node placement. These meshes are shown in figure 19. Most of these meshes contain some high aspect ratio (long and narrow) cells, despite using Delaunay triangulation to minimize these ratios. One might expect these meshes to give inaccurate results. Surprisingly, the maximum relative difference in average scalar flux calculations as compared to the rectangle calculation was still less than 0.3 percent. Top edge current for each case is contained in Table 2. Current distribution relative difference compared to rectangle calculations was also insensitive to the spatial meshes. These results are shown in figure 20.

Table 2
Top Edge Current for 0.50 Unit Random Variation in Node Placement for the Sensitivity Test Case

x	Squares	Trial #0	Trial #1	Trial #2	Trial #3	Trial #4	Trial #5
0.5	0.20780	0.20785	0.20839	0.20856	0.20822	0.20863	0.20793
1.5	0.24622	0.24632	0.24563	0.24587	0.24602	0.24594	0.24635
2.5	0.25018	0.25021	0.25035	0.25027	0.25030	0.25028	0.25034
3.5	0.24622	0.24639	0.24598	0.24602	0.24601	0.24606	0.24599
4.5	0.20780	0.20899	0.20864	0.20856	0.20812	0.20862	0.20805



Trial 1

Trial 2

Trial 3

Trial 4

Trial 5

Figure 19. Mesh Sensitivity Test Set for the Sensitivity Test Problem for 0.50 Unit Random Variation

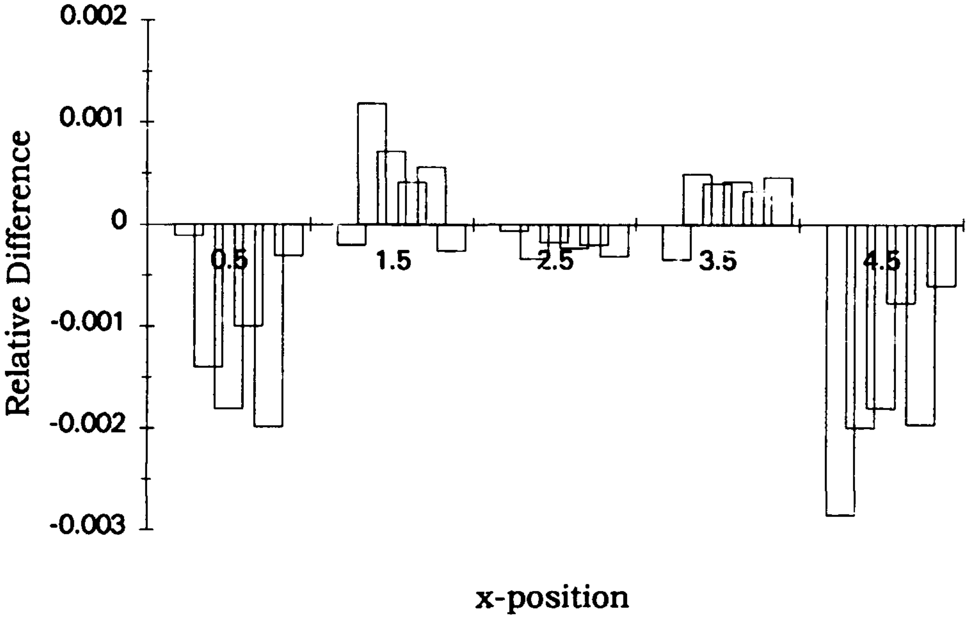


Figure 20. Top Edge Current Relative Difference for Sensitivity Problem for 0.50 Unit Random Variation in Node Placement

Modest variations in triangular mesh around one mean free path change the size and orientation of the triangle spatial cells but produce relative differences of less than 1.0 percent in top edge currents and average scalar fluxes. However, the trial 0 mesh with perfectly regular 45-degree right triangles was the worst performer. The trial 0 mesh did not produce symmetric results. However, the asymmetry is less than one might expect. Randomizing the triangle mesh appears to avoid systematic accumulation of errors and improves performance, even though some long and narrow triangles result. Randomizing the mesh is not an option with row-column rectangular meshes.

D. Test Case 1: Source Cylinder with Annular Segment Reflectors

The source cylinder with annular segment reflectors test problem is designed to show the difficulties with rectangular spatial mesh generation. This partially reflected cylinder is motivated by space power reactor designs. The test problem consists of a circular source region of radius 2.0, with scattering cross-section of 0.5, total cross-section of 0.75, and source strength of 1.0. The source region is surrounded by two concentric rings with alternating sections of void and scattering material in each ring. The scattering material has scattering cross-section of 0.5, total cross-section of 0.75, and no source. Three different configurations of this problem were observed. First is the fully closed configuration where the rings are oriented to leave no unobstructed leakage path from the source region. This configuration is shown in figure 21.

The second configuration of the partially reflected cylinder test problem is the half open configuration, which is obtained by a 45-degree rotation of the outermost ring. The half-open configuration is shown in figure 22.

The last configuration of the partially reflected cylinder problem studied is the fully open configuration, which is obtained by rotating the outermost ring an additional 45 degrees to achieve the maximum opening for leakage from the source region. The fully open configuration is shown in figure 23.

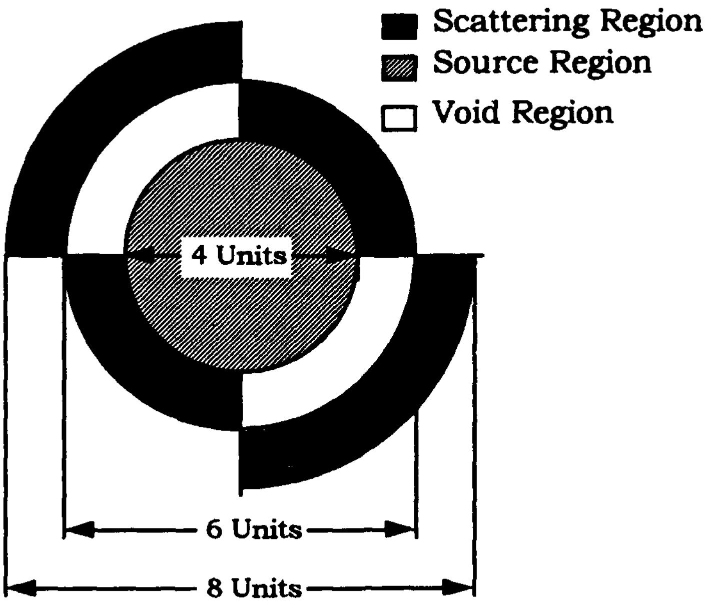


Figure 21. Fully Closed Partially Reflected Cylinder Problem

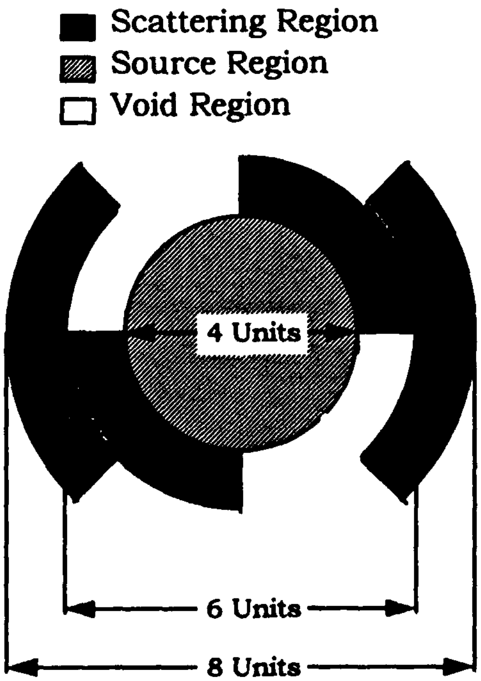


Figure 22. Half Open Partially Reflected Cylinder Problem

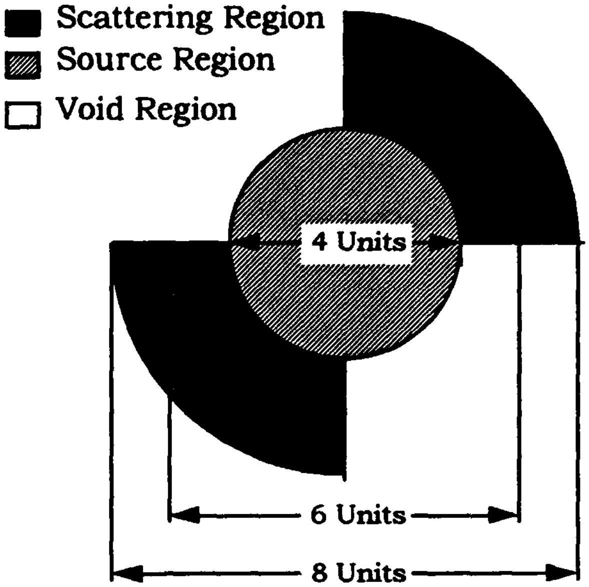


Figure 23. Fully Open Partially Reflected Cylinder Problem

1. Spatial Mesh Considerations for the Partially Reflected Cylinder Problem.

Square and triangle meshes were generated to model the problem with varying cell thicknesses. Square meshes were particularly difficult to generate because of the curved boundaries of the problem. Square meshes were generated by calculating the centroid location of each square cell and assigning the cell to the region in which the centroid was located. This scheme was applied each time the square mesh was refined. As a result, each successive mesh refinement provided a better approximation to the problem geometry.

Triangle meshes were generated by placing nodes on concentric rings and using Delaunay triangulation to construct triangles. The number of nodes on each ring was controlled by visually rejecting meshes that allowed overlap between regions of the problem. This constrained the number of nodes that could be placed on a ring.

General triangle meshes very closely modeled the problem geometry even for just a few triangles. Figure 24 shows the 80 general triangle mesh (16 cells for the source region) for the partially reflected cylinder test problem (fully open case). This mesh very closely approximates all region areas and problem boundaries. The general triangle mesh of figure 24 is contrasted by the 208 square (32 cells for the source region) mesh of figure 25. Even 208 spatial cells do not resolve the test problem geometry sufficiently to provide valuable results. The 208 square mesh does model the duct openings, but fails to conserve region areas. Table 3 shows the areas of the problem regions for the partially reflected cylinder fully open configuration for various mesh types and refinements.

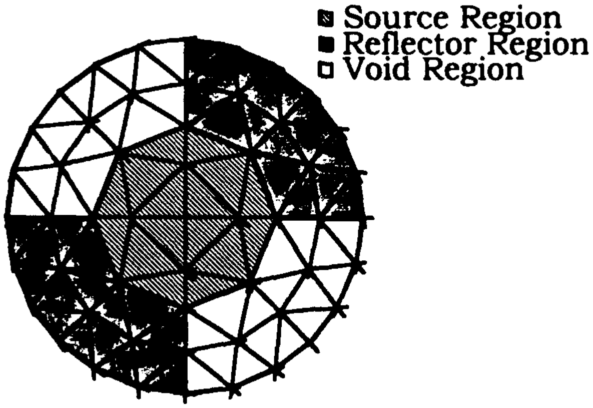


Figure 24. Fully Open Partially Reflected Cylinder Problem (80 triangle mesh)

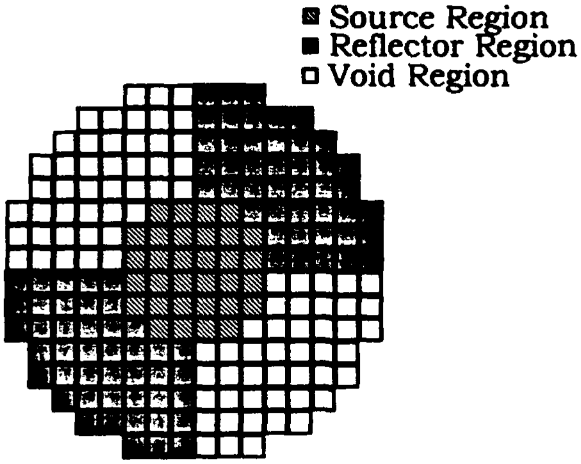


Figure 25. Fully Open Partially Reflected Cylinder Problem (208 square mesh)

Table 3
Areas for Regions of Various Meshes for the Fully Open Partially Reflected Cylinder Problem)

Type Mesh	Source	Inner Reflector	Outer Reflector
Exact	12.5664	7.854	10.9956
52 Squares	12	10	10
208 Squares	13	7.5	11
812 Squares	13	7.75	11
3228 Squares	12.6875	7.75	11
80 Triangles	12.5663	8.1437	11.0462
400 Triangles	12.5664	7.8535	10.9954
1808 Triangles	12.5664	7.8536	10.9956

Figure 26 shows the 80 triangle mesh for the half open configuration of the partially reflected cylinder test problem. Eighty triangles accurately describes the problem, very closely approximates region areas, and nearly exactly describes the vacuum openings to the source region. The 208 square cell mesh for the half-open configuration is shown in figure 27. Figure 27 shows that 208 square cells only gives the general shape of the test problem.

The fully closed 80 triangle mesh is shown in figure 28. It again models the features of the test problem very well, while distributing the materials accurately and nearly conserving their areas. The fully closed 208-cell square mesh is shown in figure 29.

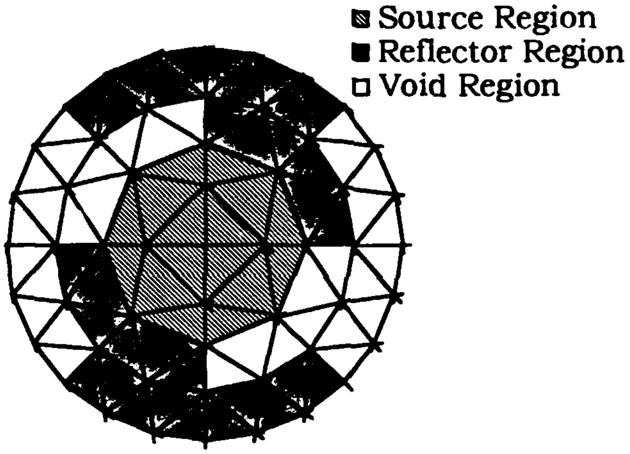


Figure 26. Half Open Partially Reflected Cylinder Problem (80 triangle mesh)

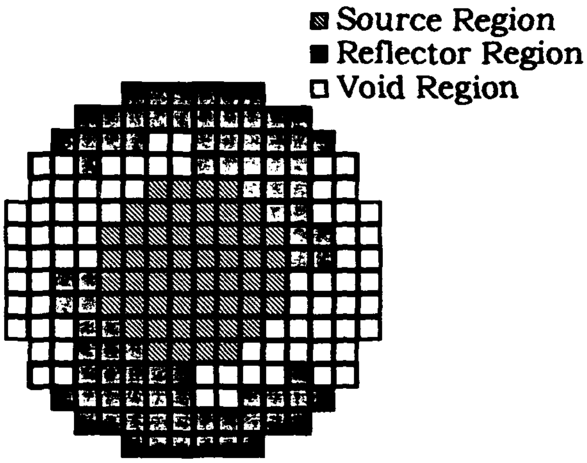


Figure 27. Half Open Partially Reflected Cylinder Problem (208 square mesh)

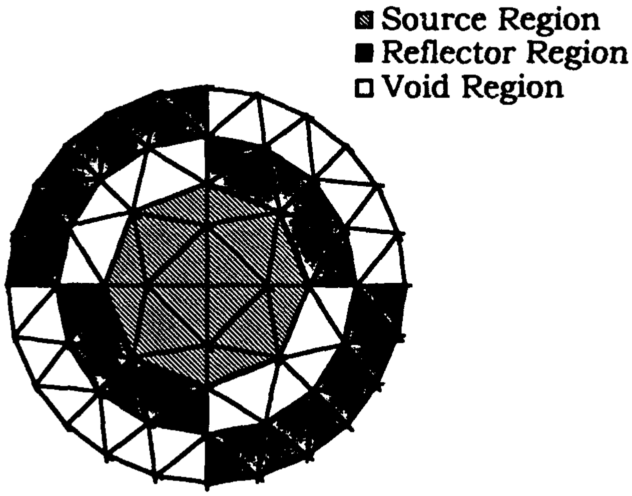


Figure 28. Fully Closed Partially Reflected Cylinder Problem (80 triangle mesh)

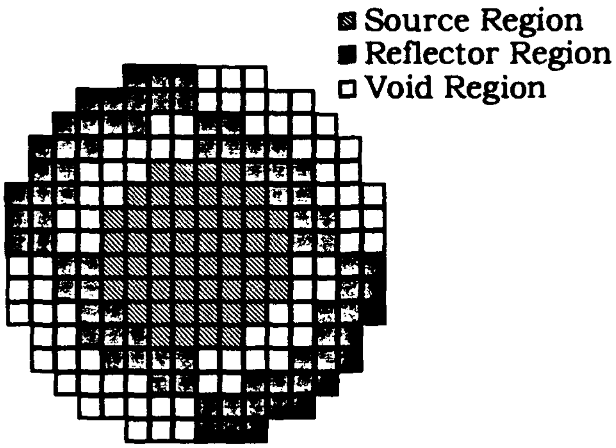


Figure 29. Fully Closed Partially Reflected Cylinder Problem (208 square mesh)

Table 4 shows the specific parameters for the meshes used in the partially reflected cylinder problem.

Table 4
Spatial Mesh Parameters for Partially Reflected Cylinder Problem

Type Mesh	Total # of Cells	Cells in Source Region	Cell Thickness (squares)	Concentric Ring Width (triangles)
Square	64	12	1.000	
Square	256	52	0.500	
Square	1024	208	0.250	
Square	4096	3228	0.125	
Triangle	80	16		1.00
Triangle	400	80		0.50
Triangle	1808	391		0.25

2. Fully Closed Configuration

Average scalar fluxes were calculated for various mesh refinements of squares and triangles. The square results were corrected for source area. The area correction was made by dividing the source region scalar flux by the ratio of the square computational grid to exact area. The results are shown in figure 30.

Figure 30 shows that a triangle mesh consisting of just 16 triangles in the source region is already within 1.0 percent of the refined square mesh average scalar flux (area uncorrected) result and by using 80 triangles the difference is reduced to less than 0.6 percent, while it takes a 52 (area uncorrected) cell square mesh to achieve 1.0 percent relative difference. Area correcting the source region produces 2.6 percent relative difference.

A better comparison of the performance of these methods are how well each successive refinement compares with the previous mesh refinement on the same type mesh. Comparing the uncorrected 208 square mesh to the uncorrected 3228 square mesh shows a relative difference of about 0.7 percent. The relative difference of corrected 208 square mesh and the 3228 corrected square mesh is about 3.4 percent. In contrast, the 80 triangle and 391 triangle meshes have a relative difference of less than 0.1 percent. Triangles are a better converged result. Correcting the square mesh source region area does not increase the performance of the square meshes, therefore only uncorrected results are presented for the remaining tests.

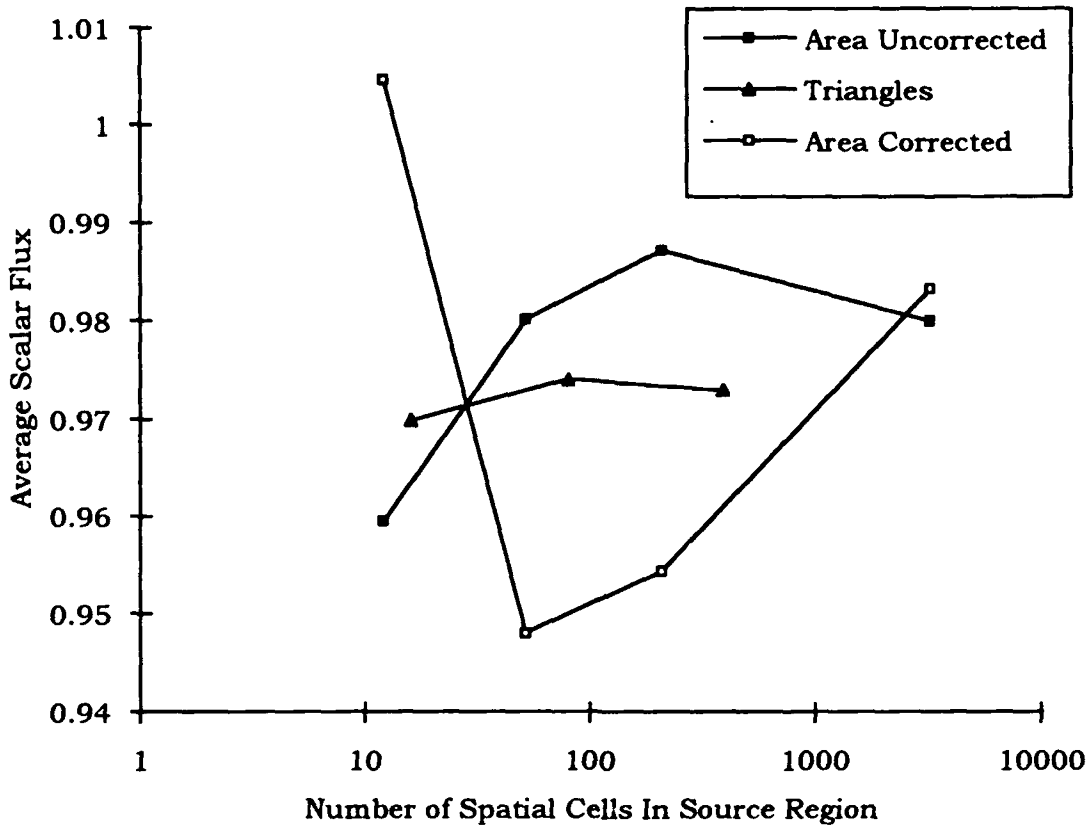


Figure 30. Fully Closed Configuration Source Average Scalar Flux

3. Half Open Configuration

The average scalar flux was calculated for various square and triangle meshes on the half open configuration for the partially reflected cylinder problem. The results are shown in figure 31.

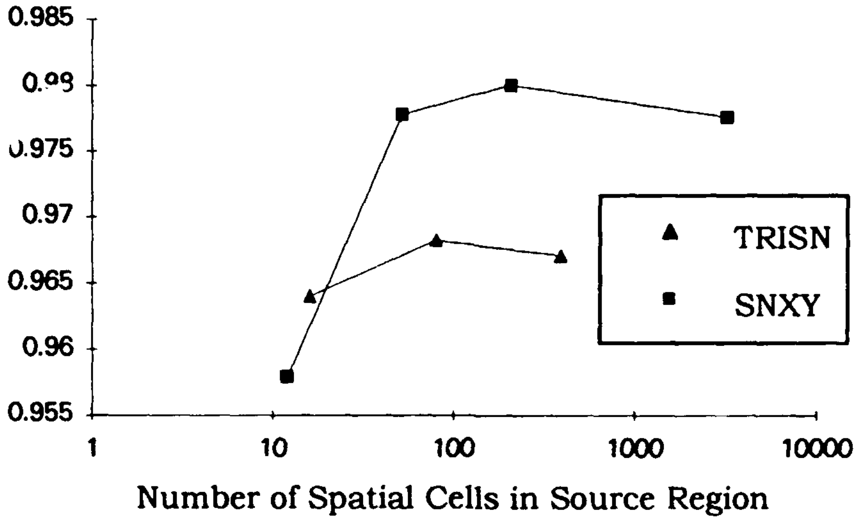


Figure 31. Half Open Configuration Source Average Scalar Flux

Figure 31 also shows that a triangle mesh consisting of just 80 triangles (16 triangles in the source) is within 0.3 percent of the refined triangle mesh average scalar flux result and using 400 triangles the difference is reduced to less than 0.01 percent. Sixty-four squares produce a relative error of 2.0 percent when compared to the 3228 square mesh, while a 1024 cell square mesh is about 0.2 percent out. Again the triangle result is better converged by a factor of 20.

4. Fully Open Configuration

The average scalar flux was calculated for various square and triangle meshes on the fully open configuration for the partially reflected cylinder problem. The results are shown in figure 32.

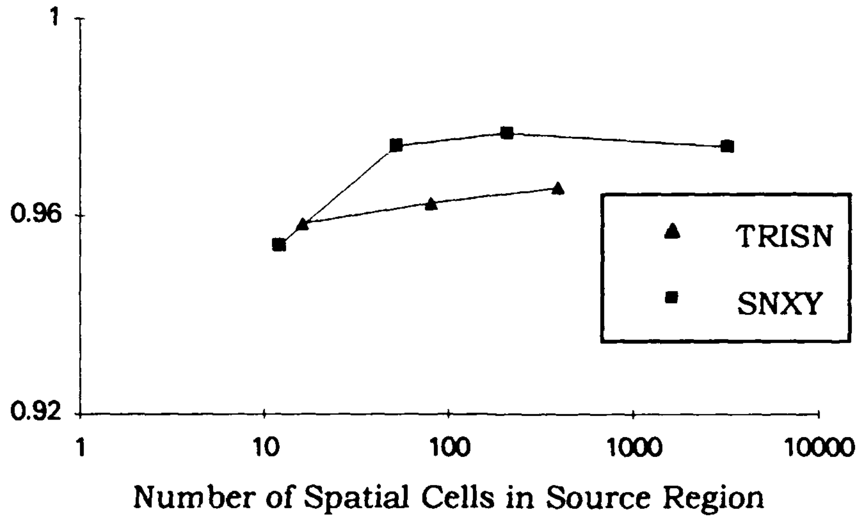


Figure 32. Fully Open Configuration Source Average Scalar Flux

Figure 32 shows that a triangle mesh consisting of just 80 triangles (16 triangles in the source region) is already within 0.8 percent of the refined triangle mesh average scalar flux result and when using 400 triangles the difference is reduced to about 0.3 percent, Using a 64 cell square mesh, the relative difference from the 3228-square mesh is about 2.4 percent. A 256-square mesh produces a relative difference from the 3228-square mesh of 0.3 percent

E. Test Case 2: The Vacuum Duct

The vacuum duct test problem is designed to demonstrate the failings of a rectangle spatial mesh, even on simple geometries. The vacuum duct problem consists of a 2 x 2 square cut along the diagonal. The two triangular pieces of this square are separated to create a vacuum duct. Two configurations are observed: the narrow duct configuration which has a vertical displacement of one unit, and the wide duct configuration, which has a vertical displacement of two units. These configurations are discussed in more detail below.

2. Narrow Vacuum Duct

The single width vacuum duct problem consists of a 2 x 3 rectangle with a vacuum duct cut through it along the rectangle diagonal. The lower triangle (source region) has a scattering cross-section of 0.5, a total cross-section of 0.75, and a source strength of 1.0. Two cases for the upper triangle (scattering region) were observed. The first case used a material with a scattering cross-section of 0.5, a total cross-section 0.75, and no source (scatter case). The second case used a pure absorber with no scattering cross-section, a total cross-section 0.75, and no source (absorber case). The narrow vacuum duct problem is shown in figure 33.

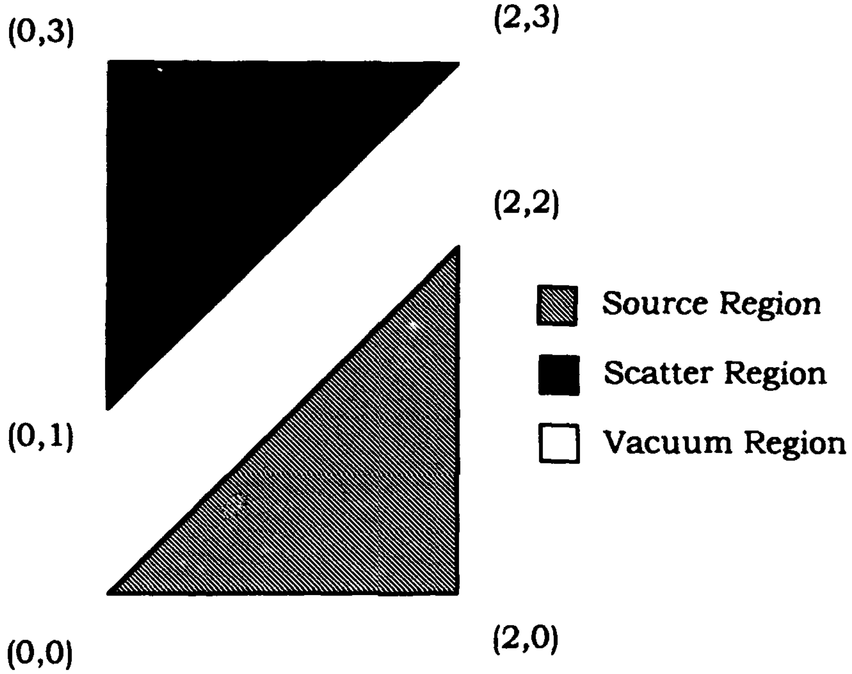


Figure 33. Narrow Vacuum Duct Problem

The narrow vacuum duct problem can be described exactly using as few as four general triangles. Using only four triangles, the source region, scattering region, and vacuum duct can be modeled exactly. The four triangle mesh is shown in figure 34. Using a six square mesh (the minimum), the areas can be modeled exactly but a different problem is described. This can be seen in figure 35. Further refinements of general triangle mesh (figure 36) refine the transport of the problem, which may not be necessary if the problem is optically thin. However, further square cell refinements are required to adequately resolve the shapes of the regions (figure 37).

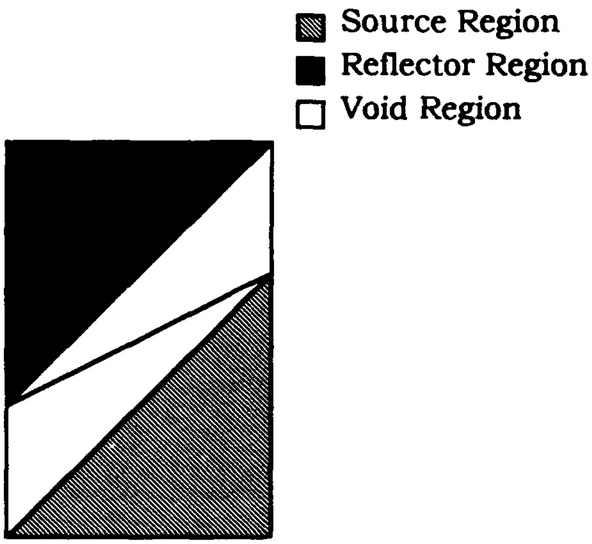


Figure 34. Narrow Vacuum Duct (4 triangle mesh)

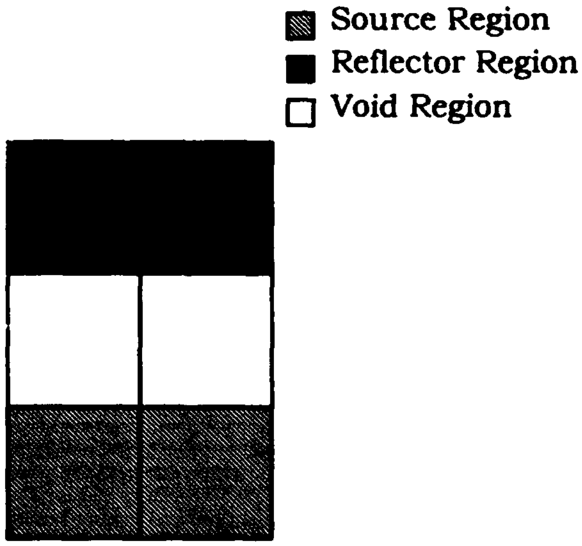


Figure 35. Narrow Vacuum Duct (6 square mesh)

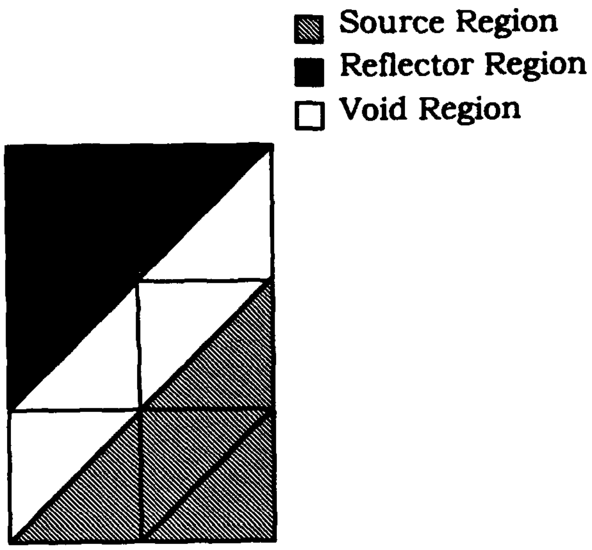


Figure 36. Narrow Vacuum Duct (12 triangle mesh)

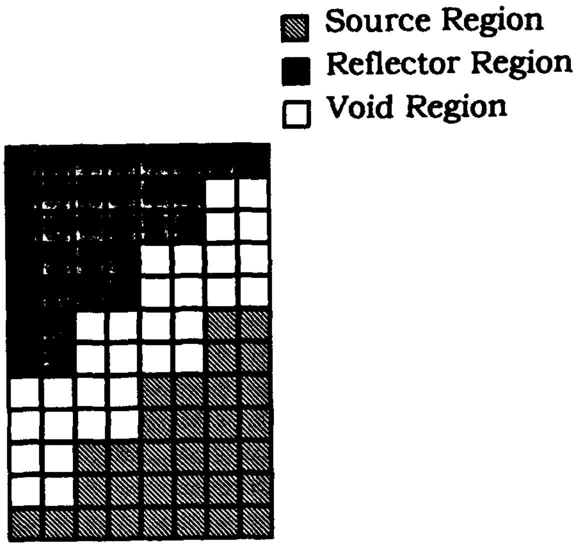


Figure 37. Narrow Vacuum Duct (96 square mesh)

Table 5 shows the specific mesh parameters for all meshes used in the narrow vacuum duct problem.

Table 5
Spatial Mesh Parameters for Narrow Vacuum Duct Problem

Type Mesh	Total # of Cells	Cells in Duct	Cells in Upper/Lower Region	Cell Thickness (squares)
Square	6	2	2	1
Square	12	4	4	1/2
Square	54	18	18	1/3
Square	96	32	32	1/4
Square	150	50	50	1/5
Square	384	128	128	1/8
Square	1536	512	512	1/16
Square	6144	2048	2048	1/32
Triangle	4	2	1	
Triangle	10	6	2	
Triangle	12	4	4	
Triangle	40	24	8	
Triangle	160	96	32	
Triangle	640	384	128	
Triangle	2560	1536	512	

Calculations were made using the linear characteristic spatial quadrature (triangle and rectangle) with S8 level symmetric angular quadrature and a convergence criterion of no more than 10^{-5} relative change in cell scalar fluxes between iterations. Average top edge currents and edge current distributions are compared for various square and triangle spatial meshes.

Table 6
Average Top Edge Current Narrow Vacuum Duct Problem (scatter case)

Type Mesh	No. of Cells	J ave Top	Rel Error
triangle	4	0.064193	0.051057
triangle	10	0.060937	0.002259
triangle	12	0.060841	0.003831
triangle	40	0.060964	0.001817
triangle	160	0.061035	0.000650
triangle	640	0.061071	6.55E-05
triangle	2560	0.061075	
square	6	0.048986	0.197937
square	12	0.056008	0.082964
square	54	0.058135	0.048138
square	96	0.059324	0.028670
square	384	0.059805	0.020794
square	1536	0.060911	0.002685
square	6144	0.061014	0.000999

The average top edge current and the relative error using the 2560 cell triangle case as a reference are listed in table 6; the relative errors are plotted in figure 38. The 2560-triangle case was used as the benchmark because the relative change in edge current between 640 and 2560 triangles was about 0.007 percent, while the relative change between 1536 and 6144 squares was still about 0.2 percent. The triangle result is the more converged result. General triangle meshes reach 1.0 percent relative error at ten triangles, while square meshes do not get to the 1.0 percent relative error level until somewhere between 384 and 1536 squares. This translates into a cell savings of greater than a factor of thirty. Figure 38 shows savings of factors of from 10 to 100 depending on the error one will tolerate.

The relative error increases when increasing the number of triangles from 10 to 12. The 12-triangle mesh has the same pattern of triangles as the trial 0 sensitivity meshes discussed earlier. This suggests that truncation errors are accumulating systematically, constituting numerical diffusion. Figure 39 shows the 10 and 12 triangle meshes. The 10 cell mesh may perform particularly well, however, because it puts more effort into refining the vacuum duct.

The general triangle linear characteristic method is in agreement with refined square calculations; also, it models the particle current distribution very well. This is seen clearly in figure 40, which shows the current distribution for various triangle grid refinements and for the most refined square mesh.

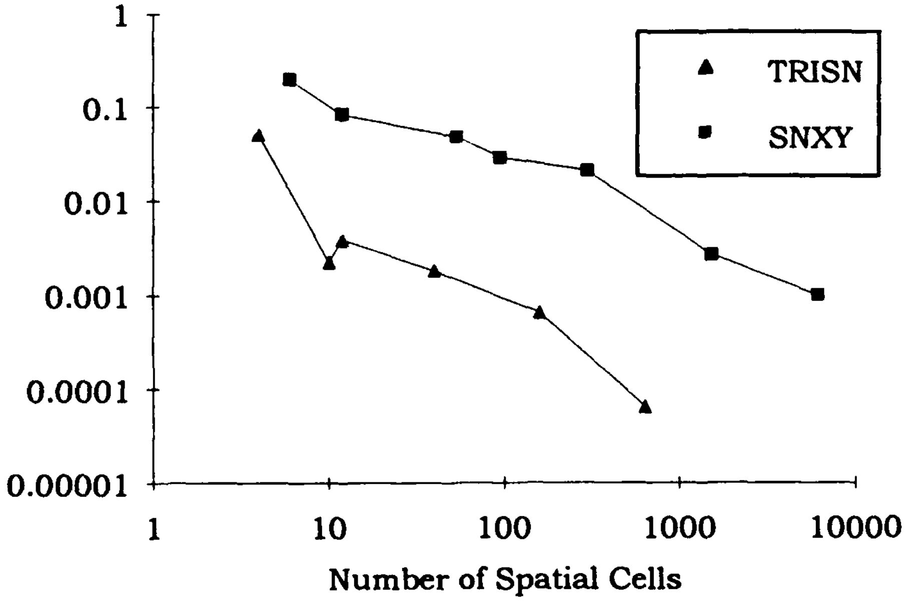


Figure 38. Relative Error Narrow Vacuum Duct Problem (scatter case)

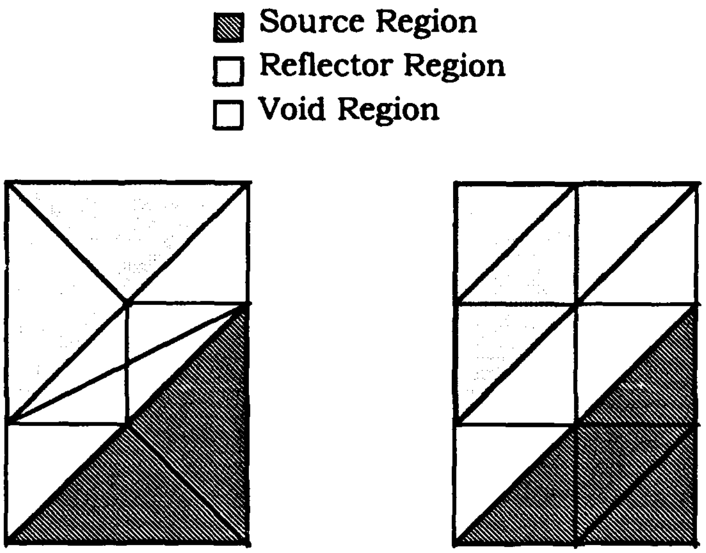


Figure 39. Narrow Vacuum Duct 10/12 Triangle Mesh Comparison

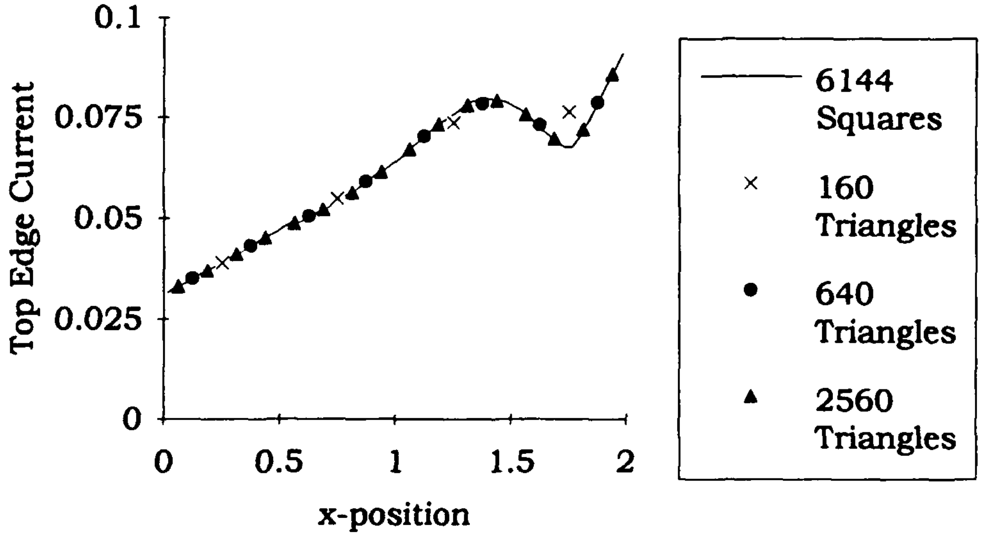


Figure 40. Top Edge Current Narrow Vacuum Duct (scatter case)

This testing was repeated using a pure absorber for the upper triangle and using the 2560 cell triangle case as a reference for calculating the relative error. Table 7 shows the results.

Table 7
Average Top Edge Current Narrow Vacuum Duct Problem
(absorber case)

Type Mesh	No. of Cells	J ave Top	Rel Error
triangle	4	0.052562	0.089029
triangle	10	0.047990	0.005698
triangle	12	0.048054	0.004372
triangle	40	0.048144	0.002507
triangle	160	0.048225	0.000829
triangle	640	0.048261	8.29E-05
triangle	2560	0.048265	
square	6	0.037157	0.230146
square	12	0.044071	0.086895
square	54	0.045728	0.052564
square	96	0.046720	0.032011
square	150	0.047148	0.023143
square	384	0.047833	0.008951
square	1536	0.048130	0.002797
square	6144	0.048217	0.000995

Table 7 shows that triangle meshes are below 1.0 percent relative error at 10 triangles, while squares do not get to the 1.0 percent relative error level until somewhere after 384 squares. This again translates to a total spatial cell savings of greater than a factor of 30. The relative error is plotted in figure 41. Figures 38 and 41 both show as fast or faster convergence for triangle linear characteristic as for rectangle linear characteristic.

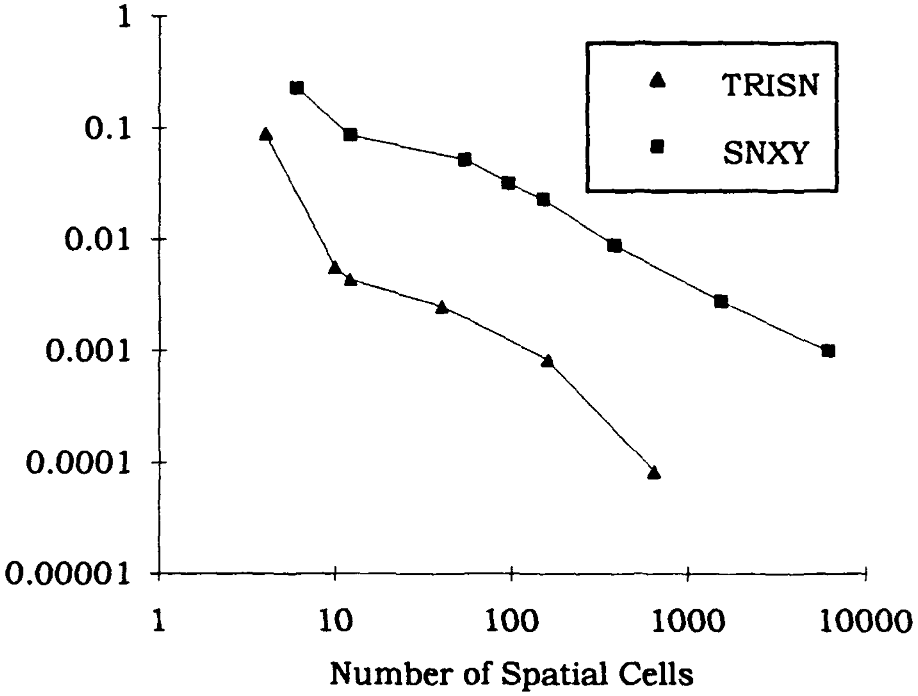


Figure 41. Relative Error Narrow Vacuum Duct Problem (absorber case)

3. Wide Vacuum Duct

The wide vacuum duct problem is a variation of the narrow vacuum duct problem where the materials remain the same but the vertical displacement of the upper triangle is doubled, widening the vacuum duct. The wide vacuum duct problem is shown in figure 42.

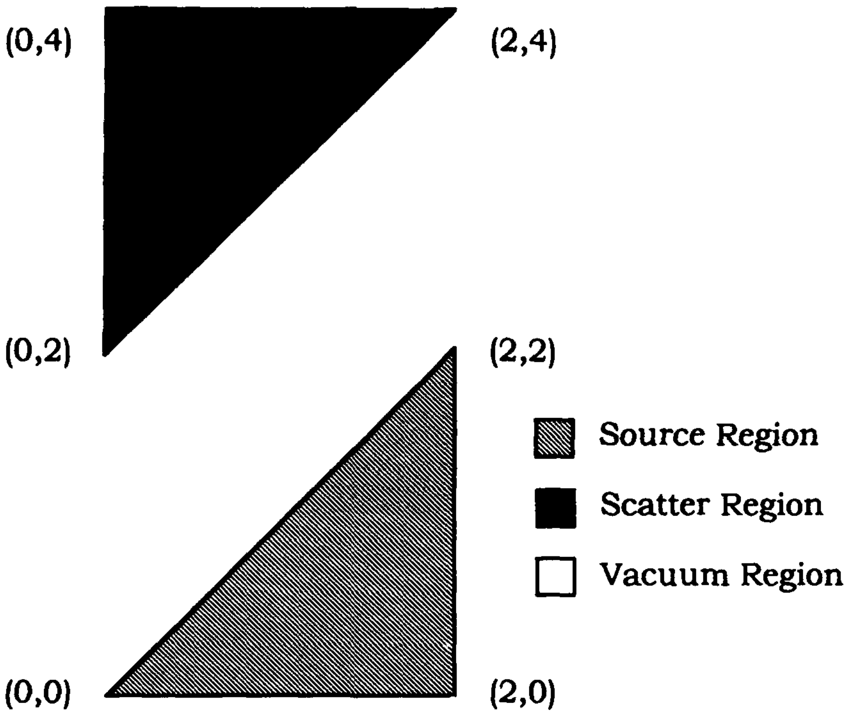


Figure 42. Wide Vacuum Duct Problem

The particle current distribution along the top edge is compared for various refinements of squares and triangles using an S8 level symmetric quadrature, linear characteristic spatial quadrature, and converging the average scalar flux to no more than 10^{-5} maximum relative change in cell scalar fluxes between iterations. Table 8 shows the average top edge current and relative error using the refined triangle case as the benchmark.

Table 8
Average Top Edge Current Wide Vacuum Duct Problem
(scatter case)

Type Mesh	No. of Cells	J ave Top	Rel Error
triangle	4	0.040227	0.005549
triangle	16	0.039804	0.005549
triangle	32	0.039873	0.003300
triangle	128	0.040000	0.000125
triangle	512	0.039995	0.000250
triangle	2048	0.040005	0.000901
square	8	0.035396	0.115211
square	32	0.037008	0.074916
square	72	0.038273	0.043295
square	128	0.039023	0.024547
square	400	0.039467	0.013448
square	512	0.039715	0.007249
square	2048	0.039907	0.002450
square	8192	0.039969	0.000900

Table 8 indicates that triangle linear characteristic has reached the 1.0 percent relative error with only 4 triangles, while square linear characteristic reaches the 1.0 percent relative error level at 512 cells. This is a total cell savings of around a factor of 250 just to resolve the problem geometry. A different rectangular cell mesh might save cells but would be complicated to generate. Comparison of the two types of meshes at the 128 cell level shows that a mesh of 128 triangle cells has a relative error of about 0.08 percent while 128 square cells gives a relative error of 2.4 percent. This translates into a factor of thirty times the error for squares. The average top edge current for various triangle and square refinements is shown in figure 43, and the corresponding errors are in figure 44.

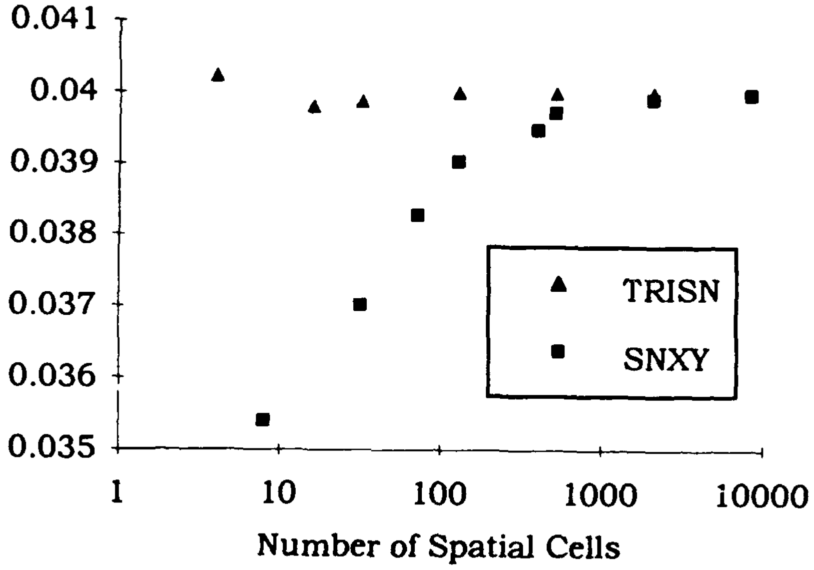


Figure 43. Top Edge Current Wide Vacuum Duct Problem (scatter case)

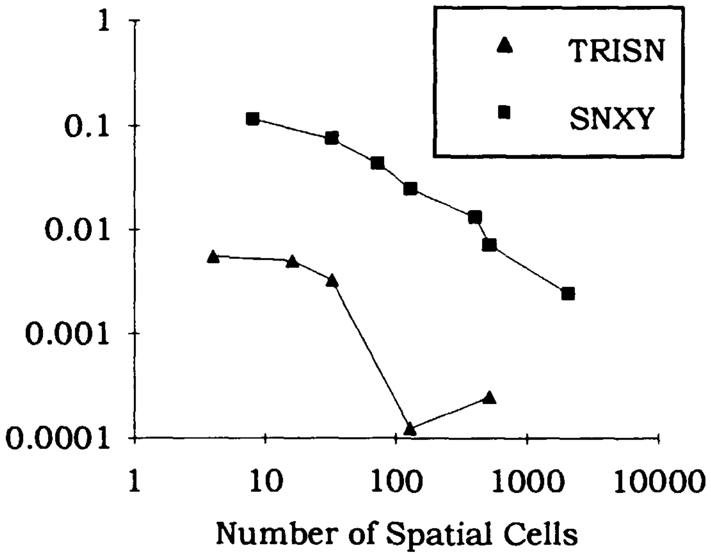


Figure 44. Relative Error Wide Vacuum Duct Problem (scatter case)

Substituting a pure absorber for the scatter material in the upper triangle, the top edge current and relative errors are compared in table 9 and figure 45.

Table 9
Average Top Edge Current Wide Vacuum Duct Problem
(absorber case)

Type Mesh	No. of Cells	J ave Top	Rel Error
triangle	4	0.032949	0.015221
triangle	16	0.032191	0.008134
triangle	32	0.032246	0.006440
triangle	128	0.032440	0.000462
triangle	512	0.032468	0.000401
triangle	2048	0.032484	
square	8	0.027562	0.150763
square	32	0.029884	0.079217
square	72	0.030807	0.050778
square	128	0.031549	0.027916
square	400	0.031980	0.014636
square	512	0.032213	0.007456
square	2048	0.032399	0.001725
square	8192	0.032455	

Table 9 shows triangle meshes reach less than 1.0 percent relative error at 16 triangle cells, while squares reach the same level at about 512 cells. At 512 cells, triangles are at about 0.05 percent relative error and squares are at 0.7 percent relative using the 8192 square cells case as the benchmark.

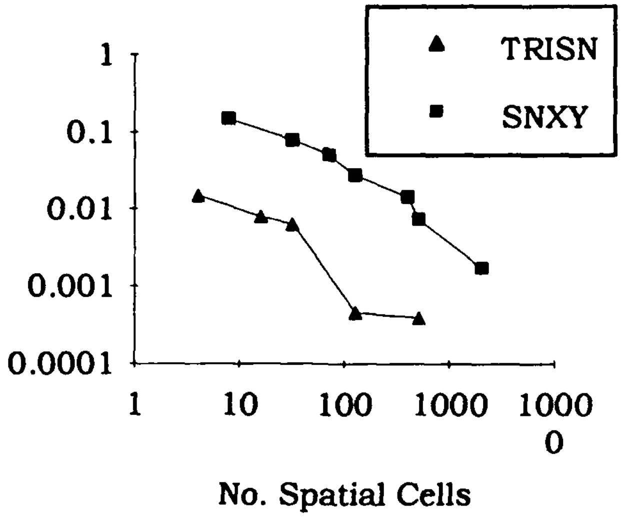


Figure 45. Top Edge Current Wide Vacuum Duct Problem
(absorber case)

F. Execution Speed

The execution speed per cell of the two codes was compared using Lathrop's problem. The square method used a 16 x 16 square mesh. The triangle method used the mesh described in figure 11, twice refined using the bisected edge refinement technique discussed earlier. This gives a total of 512 triangular spatial cells. Rectangle and triangle linear characteristic were compared using a S8 level symmetric quadrature. The problems were run on a Sun Sparc 2 workstation. For each spatial mesh, the problem was run five times (to minimize external factors, number of users, CPU load, etc.) tracking the execution time with a stop watch. All measurements were less than one percent different from the average result. The execution times were averaged for all trials on each mesh type. The average was then divided by the number of iterations to convergence (less than 10^{-5} maximum relative change in scalar flux), the number of ce!!s in the particular spatial mesh, and the number of ordinates (forty for S8 level symmetric). The triangle run time per ordinate per cell per iteration was divided by the corresponding square result to give a measure of the increased effort for a triangle cell computation. The average square cell computation time was about 1.6 x 10^{-4} second. The average triangle cell computation time was about 4.3 x 10^{-4} second, which is about 2.7 times more effort to compute a triangular cell with linear characteristic than to compute a rectangular cell. It is important to note that the square code took advantage of symmetries to precalculate many of the parameters required (e.g. exponential moment functions, optical thicknesses, etc.). The triangle code was implemented in a completely general fashion (no precalculation of parameters) to allow generalization to three-dimensions. As a result, the triangle cell computation speed could be increased.

G. Summary of Observations

Lathrop's problem validated the implementation of the general triangle linear characteristic spatial quadrature. It showed that not only was the average scalar flux calculated properly, but the average edge current and edge current distribution also were in agreement with a validated computer program. Additionally, the grid sensitivity measurements showed that average scalar flux values were insensitive (less than one percent) to triangle mesh variations for up to two-thirds of a mean free path in both x and y directions. Sensitivity measurements also showed that the trial 0 right angle meshing with triangles gave the worst performance indicating that numerical diffusion was a problem. This was also seen in right triangle meshes of Lathrop's problem and the vacuum duct problem.

Two families of test cases were investigated: the source cylinder with annular segment reflectors and the vacuum duct test problem. Each had geometric features that were difficult to model with rectangle spatial meshes. General triangle meshes much more easily modeled these geometric features. In each case, general triangle meshes very accurately modeled the geometric features of the problem with few spatial cells.

General triangle meshes provided at least a factor of three savings in the number of spatial cells required to achieve a one percent relative error level on the partially reflected cylinder test problems, while asymptotically converging at a higher rate. For some cases of the vacuum duct problems, triangles saved more than a factor of 100 in spatial cells.

Computational speed was measured to quantify the increase in effort required to use a general triangle mesh. The Lathrop problem was repeated for both the square and triangle meshes. Total time was recorded to obtain an average problem execution time. The average time of execution was divided by the total number of cells in the mesh, the number of ordinates in the angular quadrature, and the number iterations to convergence. The triangle linear characteristic method required about 2.7 times more effort than the rectangle linear characteristic method to do a cell computation for Lathrop's problem.

VI. Conclusions and Recommendations

The general triangle lin.ear characteristic spatial quadrature was developed, implemented, and tested. This required construction of a local coordinate system, new conservation relationships, and a pathway to and from coordinates systems. Additionally, data structure restrictions required development of the algorithm to support arbitrary triangle meshes.

The general triangle linear characteristic spatial quadrature provides somewhat slower convergence on Lathrop's problem as did the rectangle linear characteristic spatial quadrature. However, Lathrop's problem uses square regions and can be modeled exactly (geometrically) with a square mesh. Most real problems do not have this property. When curved region boundaries or boundaries that are not perfectly aligned with one of the cardinal (x, y) axes are introduced, rectangular spatial meshes have difficulties resolving the problem geometry. This may be the case even though the particle transport may be sufficiently resolved to provide accurate results. In fact, geometry may be the dominant factor in developing the spatial mesh for complicated problems. Triangle meshes provide an attractive alternative for these problems.

Lathrop's problem validated the derivation and implementation of the triangle linear characteristic method by showing convergence of average scalar flux, top edge average current, and the distribution of top edge current to results from a previously validated rectangle linear characteristic and diamond difference code.

Grid sensitivity measurements showed that angular flux and particle current were relatively insensitive to randomized triangle meshes even when meshing produced high aspect ratio triangles.

Two families of test cases were developed which demonstrated some advantages of the new triangle linear characteristic method. They were the source cylinder with annular segment reflectors problems and the vacuum duct test problems. For each test problem, triangle meshes very accurately modeled the geometry of the problem with few cells, while rectangle meshes required many cells to adequately model the problem. Triangle meshes were consistently more accurate than square meshes for the partially reflected cylinder test problems. For the vacuum duct problems, the triangle method achieved the same accuracy as the square method, but with 3 to more than a 100 times fewer triangles than squares. Triangle meshes converged more rapidly than did square meshes for these problems, obtaining accurate results with only a few cells. Spatial cell savings were observed even though the problems were optically thin, demonstrating that geometric considerations can drive the need for mesh refinement. Because rectangular spatial cells poorly resolve inclined or curved surfaces, convergence to accurate results is slowed when such surfaces are present. General triangles accurately approximate these surfaces with many fewer cells. The savings in spatial cells can allow more complicated problems to be attempted.

Triangle meshes that contained only right triangles gave less accurate results than did arbitrary meshes in all instances tested. This is due to the fact that truncation errors accumulate systematically due to the repetitive nature of the mesh. These errors can readily be minimized with randomized triangle meshes. Rectangular meshes do not have this flexibility.

There are several potential areas for extension of this work. The TRISN computer code is a one-energy group, time independent, isotropic scatter code which is sufficient to test new spatial quadratures. However, expansion of the code to model more realistic problems would be extremely useful.

During this effort, the ability to generate general triangle meshes was extremely limited. The Delaunay triangulation algorithm has particular difficulty with curved boundaries shared between regions. (Such boundaries are concave with respect to one of the regions, and the Delaunay triangulation algorithm often specifies triangles that lie outside such a boundary, and thus are in the wrong region). Development or adaptation of an existing general triangle meshing algorithm would provide the ability to perform calculations on more robust problems. The TRISN algorithm could be modified to handle boundary conditions other than vacuum boundaries. This would involve designing and building angular quadratures that would have reflection symmetry at every boundary. This capability would allow problems with off-axis symmetries to be modeled with fewer spatial cells. Extension of the case 0 spatial quadrature development to other spatial quadratures would be valuable. Possibly implementing a case 0 linear nodal method or a case 0 exponential characteristic spatial quadrature would be very interesting. This could allow thicker cells to be used.

Finally, many interesting problems have regions of very high scatter. A useful extension of this work would be to investigate convergence acceleration techniques and their interaction with the general triangle discrete ordinates algorithm.

Bibliography

1. Abu Schmumays, I. K., "Compatible Product Quadratures for Neutron Transport in X-Y Geometry, " Nuclear Science and Engineering, 64, 299-316 (1977).

2. Alcouffe R.E., E. W. Larsen, W. F. Miller, Jr., and B. R. Wienke, "Computational Efficiency of Numerical Methods for the Multigroup, Discrete-Ordinates Neutron Transport Equations: The Slab Geometry Case, "Nuclear Science and Engineering, 71, 111-127 (1979)

3. Alcouffe R. , and W.F. Larsen, "A Review of Characteristic Methods used to Solve the Linear Transport Equation", Proceedings of the American Nuclear Society Topical Meeting on Advances in Mathematical Methods for the Solution of Engineering Problems, Munich FRG, 27-29 April 1981, Vol. 1:3-16 (1981)

4. Carson, B. G., and G. I. Bell, "Solution of the Transport Equation by the Sn Method." Proceedings of the Second United Nations Conference on Peaceful Uses of Atomic Energy, 16, 535 (1958)

5. Larsen, E. W. and R. E. Alcouffe. "The Linear Characteristic Method for Spatially Discretizing the Discrete-Ordinates Equations in x-y Geometry," Proceedings of the American Nuclear Society Topical Meeting on Advances in Mathematical Methods for the Solution of Engineering Problems, Munich, FRG, April 27-29, 1981, Vol. 1:99-113.

6. Larsen, Edward W. and Warren F. Miller Jr., "Convergence Rates of Spatial Difference Equations for the Discrete-Ordinates Neutron Transport Equations in Slab Geometry," Nuclear Science and Engineering, 73, 76-83 (1980)

7. Lathrop, K. D., "Remedies for Ray Effects," Nuclear Science and Engineering, 45, 255-268 (1971).

8. Lathrop, K. D. "Spatial Differencing of the Transport Equation: Positivity vs. Accuracy," Journal of Computational Physics, 4: 475-498 (1971)

9. Lathrop K. D., and F. W. Brinkley Jr., "The Theory and Use of the General Geometry TWOTRAN Program", LA-9432, Los Alamos Scientific Laboratory (1973).

10. Lewis, E. E. and W. F. Miller, Jr., Computational Methods of Neutron Transport, John Wiley and Sons, New York, NY (1984).

11. Mathews, Kirk A., "Discrete Elements Method of Neutral Particle Transport", Ph.D. Dissertations AFIT/DS/83-5. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH. October 1983.

12. Mathews, Kirk A., "Adaptive Characteristic Spatial Quadratures for Discrete Ordinates Neutral Particle Transport -- The Slab Geometry Case," Transport Theory and Statistical Physics 19(6): 419-458 (1990).

13. Mathews Kirk A. and Bryan M. Minor, "Step Adaptive Characteristic Spatial Quadrature for Discrete Ordinates Neutral Particle Transport in Two-Dimensional Cartesian Coordinates.", Proceedings of the American Nuclear Society International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics, Pittsburgh, PA, 28 Apr 02 1991 - May 1991, Vol 3, 13.2.4-1 - 13.2.4-12 (1991).

14. Paternoster, Richard R., "A Linear Characteristic-Nodal Transport Method for the Two-Dimensional (X,Y)-Geometry Multigroup Discrete Ordinates Equations over an Arbitrary Triangle Mesh," LA-10092-T, Los Alamos National Laboratory, (1984).

15. Rhodes, W. A., and F. R. Mynatt, "The DOT-II Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280, Oak Ridge National Laboratory (1973).

16. Walters, Wallace F., "Augmented Weighted-Diamond Form of the Linear Nodal Scheme for Cartesian Coordinate Systems", Nuclear Science and Engineering, 92, 192-196 (1986)

17. Walters, Wallace F., "The Relationship between Finite Elements Methods and Nodal Methods in Transport Theory", Progress in Nuclear Energy, 18, 21-26 (1986)

18. Walters, Wallace F. and R. Douglas O'Dell, "Nodal Methods for Discrete-Ordinates Transport Problems in (X, Y) Geometry", Proceedings of the American Nuclear Society Topical Meeting on Advances in Mathematical Methods for the Solution of Engineering Problems, Munich FRG, 27-29 April 1981, Vol. 1:115-129 (1981)

Vita

Captain Dennis J. Miller was born on 17 September 1957 in Seattle, Washington. He graduated from A. C. Davis Senior High School, Yakima, Washington in 1975. He enlisted in the United States Air Force in 1976. After basic training, he had assignments at the 410th Bombardment Wing, K. I. Sawyer AFB, Michigan and then at the 92nd Bombardment Wing, Fairchild AFB, Washington where he served as an aircraft maintenance crew chief and technician. He was accepted to the Airman Education and Commissioning Program (AECP) in 1983 and entered the University of Washington that fall. He received a Bachelor of Science in Engineering (Nuclear Emphasis) degree from the University of Washington in 1985. He received his commission from Officer Training School as a Second Lieutenant in the United States Air Force in the spring of 1986. After commissioning, he was assigned to Air Force Logistics Command (AFLC), Tinker AFB, Oklahoma, where he was assigned to OC-ALC/MMEAS as a nuclear survivability/vulnerability engineer. He entered the Nuclear Engineering program at the Air Force Institute of Technology (AFIT) in 1988 and graduated with a Master of Science in Nuclear Engineering degree in the spring of 1990. He was accepted to the AFIT Doctoral program in 1990.

Permanent Address: 1216 Rock Avenue
Yakima, WA 98902

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1993

3. REPORT TYPE AND DATES COVERED
Dissertation

4. TITLE AND SUBTITLE
LINEAR CHARACTERISTIC SPATIAL QUADRATURE FOR DISCRETE ORDINATES NEUTRAL PARTICLE TRANSPORT ON ARBITRARY TRIANGLES

5. FUNDING NUMBERS

6. AUTHOR(S)
Dennis J. Miller, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
A' 'DS/ENP/93-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A new discrete ordinates spatial quadrature for arbitrary triangular cells is derived and compared to its rectangular cell linear characteristic counterpart. The triangular mesh is more flexible, allowing curved surfaces and off-axis angles to be approximated with many fewer spatial cells. The triangle method is consistently more accurate on example problems tested here. Arbitrary orientation and size of the triangles allow non-patterned meshes to be developed which appears to ameliorate numerical diffusion. The triangle linear characteristic quadrature converges at nearly the same rate as rectangular linear characteristic on Lathrop's problem. Mesh sensitivity measurements show large variations in triangle vertex locations produce less than 1.0 percent variation in results. Test cases included a rectangular region with diagonal vacuum duct, and cylindrical source region with rotated rings of annular segmented reflectors. The triangle linear characteristic quadrature is more cost effective on these problems achieving a relative error of less than 1.0 percent with a factor of three to over a hundred fewer spatial cells, with less than three times the computational cost per cell. This spatial cell savings should increase the practical problem domain for which discrete ordinates is usable.

14. SUBJECT TERMS
Neutron Transport, Photon Transport, Boltzmann Equation, Numerical Methods

15. NUMBER OF PAGES
134

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL